

Seismic Velocity Estimation from Time Migration

by

Maria Kourkina Cameron

Diplom (Moscow Institute of Physics and Technology) 1998

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Applied Mathematics

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor James Sethian, Chair
Professor Alexandre Chorin
Professor Dung-Hai Lee

Spring 2007

Seismic Velocity Estimation from Time Migration

Copyright © 2007

by

Maria Kourkina Cameron

Abstract

Seismic Velocity Estimation from Time Migration

by

Maria Kourkina Cameron

Doctor of Philosophy in Applied Mathematics

University of California, Berkeley

Professor James Sethian, Chair

This is concerned with imaging and wave propagation in nonhomogeneous media, and includes a collection of computational techniques, such as level set methods with material transport, Dijkstra-like Hamilton-Jacobi solvers for first arrival Eikonal equations and techniques for data smoothing. The theoretical components include aspects of seismic ray theory, and the results rely on careful comparison with experiment and incorporation as input into large production-style geophysical processing codes.

Producing an accurate image of the Earth's interior is a challenging aspect of oil recovery and earthquake analysis. The ultimate computational goal, which is to accurately produce a detailed interior map of the Earth's makeup on the basis of external soundings and measurements, is currently out of reach for several reasons. First, although vast amounts

of data have been obtained in some regions, this has not been done uniformly, and the data contain noise and artifacts. Simply sifting through the data is a massive computational job. Second, the fundamental inverse problem, namely to deduce the local sound speeds of the earth that give rise to measured reflected signals, is exceedingly difficult: shadow zones and complex structures can make for ill-posed problems, and require vast computational resources.

Nonetheless, seismic imaging is a crucial part of the oil and gas industry. Typically, one makes assumptions about the earth's substructure (such as laterally homogeneous layering), and then uses this model as input to an iterative procedure to build perturbations that more closely satisfy the measured data. Such models often break down when the material substructure is significantly complex: not surprisingly, this is often where the most interesting geological features lie.

Data often come in a particular, somewhat non-physical coordinate system, known as time migration coordinates. The construction of substructure models from these data is less and less reliable as the earth becomes horizontally nonconstant. Even mild lateral velocity variations can significantly distort subsurface structures on the time migrated images. Conversely, depth migration provides the potential for more accurate reconstructions, since it can handle significant lateral variations. However, this approach requires good input data, known as a "velocity model".

We address the problem of estimating seismic velocities inside the earth, i.e., the problem of constructing a velocity model, which is necessary for obtaining seismic images in regular Cartesian coordinates. The main goals are to develop algorithms to convert time-migration velocities to true seismic velocities, and to convert time-migrated images to depth images in regular Cartesian coordinates.

Our main results are three-fold. First, we establish a theoretical relation between the true seismic velocities and the "time migration velocities" using the paraxial ray tracing. Second, we formulate an appropriate inverse problem describing the relation between time

migration velocities and depth velocities, and show that this problem is mathematically ill-posed, i.e., unstable to small perturbations.

Third, we develop numerical algorithms to solve regularized versions of these equations which can be used to recover smoothed velocity variations. Our algorithms consist of efficient time-to-depth conversion algorithms, based on Dijkstra-like Fast Marching Methods, as well as level set and ray tracing algorithms for transforming Dix velocities into seismic velocities. Our algorithms are applied to both two-dimensional and three-dimensional problems, and we test them on a collection of both synthetic examples and field data.

Professor James Sethian
Dissertation Committee Chair

Contents

Contents	i
List of Figures	iv
List of Tables	vi
Acknowledgements	vii
1 Introduction	1
1.1 Overview and summary of the goals and results	1
1.2 Background: settings and terminology	4
1.2.1 Seismic images	4
1.2.2 High frequency approximation	8
1.3 Time migration coordinates and image rays	9
1.4 Travel time approximation	11
1.5 Emerging wave front	12
1.5.1 General 3D case	12
1.5.2 2D simplification	13
1.6 Dix inversion	14
2 Forward modeling of time migration velocities	16
2.1 Paraxial ray tracing	16
2.2 Relation between the matrix \mathbf{K} and the true seismic velocities in 3D	19
2.3 Relation between the Dix velocities and the true seismic velocities in 2D . .	23
2.4 Analytical example	24
2.5 Statement of the inverse problem	32

2.5.1	The inverse problem in 2D	33
2.5.2	The inverse problem in 3D	36
3	Numerical algorithms in 2D	38
3.1	Efficient time-to-depth conversion algorithm	38
3.1.1	Eulerian formulation of the boundary value problem	39
3.1.2	Numerical algorithm	40
3.1.3	Causality	44
3.1.4	Boundary effects	45
3.1.5	Synthetic data examples	46
3.2	Algorithms producing the seismic velocities from the migration velocities . .	52
3.2.1	Ray tracing approach	52
3.2.2	Level set approach	55
4	Synthetic data examples in 2D	58
4.1	Example 1	58
4.2	Example 2	60
4.3	Example 3	61
5	Field data examples	63
6	Numerical algorithms in 3D	67
6.1	Step 1: Ray tracing algorithm	68
6.1.1	The ray equation in the spherical coordinates	68
6.1.2	The ray tracing algorithm	71
6.2	Step 2: Recomputation of the velocity using the found image rays	74
6.3	Step 3: Time-to-depth conversion algorithm	74
7	Synthetic data examples in 3D	77
7.0.1	Example 1	77
7.0.2	Example 2	80
7.1	Example 3	80
8	Summary and future work	84
8.1	Summary	84
8.2	Future work	85

8.2.1	Causality issue in the time-to-depth conversion algorithm	85
8.2.2	Field data example in 3D	86
8.2.3	Level set approach in 3D	86
8.2.4	First and second derivatives estimation for noisy data	86
Bibliography		87

List of Figures

1.1	The main idea of this work.	2
1.2	The approaches and the algorithms: (a) 2D, (b) 3D.	4
1.3	The time coordinates (left), and the depth coordinates (right).	5
1.4	An example of a time migrated image.	7
1.5	Image rays and time migration coordinates.	10
1.6	Travel time approximation.	11
1.7	Emerging wave front.	12
1.8	Dix inversion.	14
2.1	Paraxial ray tracing.	17
2.2	Illustration for Theorem 1.	20
2.3	Illustration for the analytical example.	26
2.4	Illustration for the analytical example.	28
2.5	Illustration for the analytical example.	31
3.1	Section 3.1.1. Relation between (x, z) , x_0 and T	39
3.2	Fast Marching Method. Black, grey and white dots represent "Accepted", "Considered" and "Unknown" points respectively.	41
3.3	(a): The exact velocity $v(x, z) = 1 + \frac{1}{2} \cos \frac{\pi x}{3} \sin \frac{\pi z}{3}$; (b): the input data $v(x_0, T)$; (c): the found velocity $v(x, z)$; (d) the relative error: its maximus is less than 5 percent.	47
3.4	The image rays computed for the exact velocity.	48
3.5	(a): The exact velocity $v(x, z)$; (b): the input data $v(x_0, t_0)$; (c): the found velocity $v(x, z)$; (d) the relative error and the image rays.	49
3.6	(a): The exact velocity $v(x, z)$; (b): the input data $v(x_0, t_0)$; (c): the found velocity $v(x, z)$; (d) the relative error and the image rays.	51

4.1	(a): the exact velocity $v(x, z)$; (b) the image rays; (c): the input data $f(x_0, t) \equiv v_{Dix}(x_0, t)$; (d): the found velocity $v(x, z)$	59
4.2	(a): the exact velocity $v(x, z)$; (b) the Dix velocity converted to depth by "vertical stretch"; (c): the found velocity $v(x, z)$ and the image rays. . . .	61
4.3	(a): The exact velocity $v(x, z)$; (b): the Dix velocity converted to depth; (c): the found velocity $v(x, z)$ and the image rays.	62
5.1	Left: seismic image from North Sea obtained by prestack time migration using velocity continuation Fomel [2003]. Right: the corresponding time migration velocity.	63
5.2	The found seismic velocity $v(x, z)$ and the image rays computed from it. . .	64
5.3	The smoothed Dix velocity $v_{Dix}(x_0, t_0)$ (left) vs the found seismic velocity $v(x, z)$ (right).	64
5.4	(a) The poststack depth migrated image obtained with the found $v(x, z)$; (b) The prestack time migrated image converted to depth.	65
5.5	The poststack depth migration using the Dix velocities (left) vs the poststack depth migration using the estimated seismic velocities (right).	66
7.1	Example 1. (a) The exact velocity; (b) the velocity found by our ray tracing approach; (c) the heuristic estimate estimate analogous to the Dix inversion, converted to depth; (d) the image rays projected onto the earth surface. . .	79
7.2	Example 2. (a) The exact velocity; (b) the velocity found by our ray tracing approach; (c) the heuristic estimate estimate analogous to the Dix inversion, converted to depth; (d) the image rays projected onto the earth surface. . .	81
7.3	Example 3. (a) The exact velocity; (b) the velocity found by our ray tracing approach; (c) the heuristic estimate estimate analogous to the Dix inversion, converted to depth; (d) the image rays projected onto the earth surface. . .	83

List of Tables

1.1	Comparison of time migration and depth migration	6
4.1	The maximal relative errors produced by the time-to-depth conversion, the ray tracing and the level set algorithms on the data from the velocity field (4.1).	60

Acknowledgements

First of all, I want to express my deepest gratitude to Professor Alexandre Chorin who made the most important events in my life my possible. First, he got me admitted to UC Berkeley, by going somewhere at the crucial moment and yelling at someone important to convince that my English would not be a problem. Second, he raised for me a very kind, smart and handsome guy Chris Cameron (truly a prince!) and kept him in the graduate school long enough so that our terms here overlapped for one year.

I want to express my deepest gratitude to my advisor Professor James Sethian for being the best thesis advisor I can fantasize. He always was very encouraging, gave me very valuable advice about how to deal with difficulties and had a lot of patience to correct my writings. Doing research with him was always very enjoyable.

I am very grateful to Dr. Sergey Fomel, who contributed a lot to the present work.

I am cordially thanking my grandmother Lidia Bashlykova who helped me to take care of my first one, and then two children and made me able to do research and write thesis. I am very grateful to my mother, Dr. Elena Kurkina, who twice saved me by coming in critical situations from Russia to US (one time in three days upon my request) to help me with my daughter and let me work.

I am very grateful to my husband Dr. Chris Cameron for always being very supportive and understanding. Who had patience to listen to my talk rehearsals as many times as necessary, to read and correct my mathematical compositions written in English, to solve my numerous problems with software and always stay nice and sweet.

Curriculum Vitæ

Maria Kourkina Cameron

Personal

Born April 25, 1976, Moscow, Russia

Education

2007 University of California, Berkeley

Ph.D., Applied Mathematics

1998 Moscow Institute of Physics and Technology, Russia

M.S., Applied Mathematics

Research Interests

Scientific Computing, PDE, Computational Physics, Geophysics,
Numerical Analysis

Publications

1 *Seismic velocity estimation and time-to-depth conversion of time-migrated images,*

Maria Cameron*, UC Berkeley; Sergey Fomel, UT Austin; James Sethian, UC Berkeley,

SEG/New Orleans 2006 Technical Program Online (SVIP 1.7)

<http://abstracts.seg.org/techprog.cfm?pMeetingID=3>

2 *Seismic Velocity Estimation from Time Migrated Velocities,*

Cameron, M. K., Fomel, S. B., Sethian, J. A., Inverse Problems,
2006, Submitted

Time to depth conversion and seismic velocity estimation using time migration velocities, Maria Cameron, Sergey Fomel, James Sethian., Geophysics, Letters, 2007, Submitted.

Chapter 1

Introduction

1.1 Overview and summary of the goals and results

Seismic data are the records of the sound wave amplitudes P described by the wave equation

$$\Delta P(x, y, z; t) = \frac{1}{v^2(x, y, z)} \frac{\partial^2}{\partial t^2} P(x, y, z; t). \quad (1.1)$$

where $v(x, y, z)$ is speed of propagation of the waves in the earth. In this work, we consider only the seismic data coming from the acoustic P waves and refer to $v(x, y, z)$ as the *seismic velocity*. This velocity is typically unknown, and its determination is the subject of the present work.

One common fast and robust process of obtaining seismic images is called *time migration* (see e.g. Yilmaz [2001]). This process is considered adequate for the areas with mild lateral velocity variation, i.e. where v depends mostly on z and only slightly on x and y . However, even mild lateral velocity variations can significantly distort subsurface structures on the time migrated images. Moreover, time migration produces images in very specific time migration coordinates (\mathbf{x}_0, t_0) (explained below), and the relation between them and the Cartesian coordinates can be nontrivial if the velocity varies laterally.

One "side product" of time migration is mean velocities $\mathbf{v}_m(\mathbf{x}_0, t_0)$, known as *time migration velocities*. We will refer to them as *migration velocities* for brevity. In the case

where the seismic velocity depends only on the depth, these velocities are close to the root-mean-square (RMS) velocities Dix [1955]. In the general case, these velocities relate to the radius of curvature of the emerging wave front Hubral and Krey [1980].

An alternative approach to obtaining seismic images is called *depth migration* Yilmaz [2001]. This approach is adequate for areas with lateral velocity variation, and produces seismic images in regular Cartesian coordinates. The major problem with this approach is that its implementation requires the construction of a *velocity model* for the seismic velocity $v(x, y, z)$. It can be both difficult and time consuming to construct an adequate velocity model: an iterative approach of guesswork followed by correction is often employed.

The main idea of this work is to construct a velocity model $v(\mathbf{x})$ from the migration velocities given in the time migration coordinates (\mathbf{x}_0, t_0) (see a block-scheme in Fig. 1.1) . Using these velocities one can then perform depth migration to obtain an improved seismic image in the Cartesian coordinates \mathbf{x} . As an alternative to depth migration, one can instead directly convert a time migrated image to "depth" (regular Cartesian coordinates) using the additional outputs of our construction $\mathbf{x}_0(\mathbf{x})$ and $t_0(\mathbf{x})$.

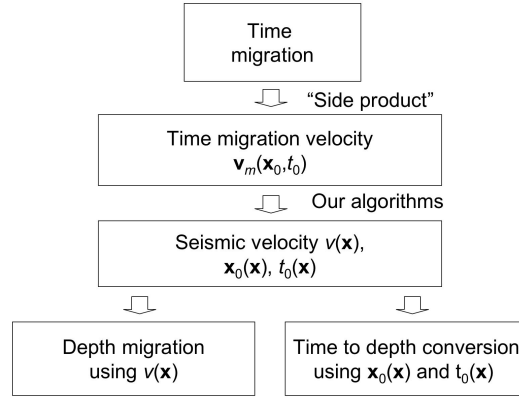


Figure 1.1. The main idea of this work.

Thus, our goals are to create fast and robust algorithms to:

1. Convert the migration velocities $\mathbf{v}_m(\mathbf{x}_0, t_0)$ to the true seismic velocities $v(\mathbf{x})$;
2. Convert time migrated images (in (\mathbf{x}_0, t_0) coordinates) to "depth" (to images in regular Cartesian coordinates \mathbf{x}).

The end result is to construct more accurate seismic images cheaply and routinely. Our results are the following:

1. We begin by producing theoretical relations between the migration velocity and the true seismic velocity in 2D and 3D.
 - In 2D the Dix velocities $v_{Dix}(x_0, t_0)$ which are a conventional estimate of true seismic velocities from the migration velocities, can be used instead of the migration velocities as a more convenient input.
 - The input data in the 3D case are a bit different. Time migration can be performed in such a way that a set of certain 2×2 matrices $\mathbf{K}(x_0, y_0, t_0)$ is determined. These matrices divided by the time t_0 have dimension of the velocity squared, and we can call the entry-wise square roots of them migration velocities. They can be converted into matrices $\mathbf{F}(x_0, y_0, t_0)$ which we use as an input for our 3D numerics.
2. Next, we formulate an appropriate inverse problem describing the relation between time migration velocities and seismic velocities, and show that this problem is mathematically ill-posed, i.e., unstable to small perturbations.
3. We develop numerical algorithms to solve the regularized versions of this problem in 2D and 3D which can be used to recover smoothed velocity variations. Our algorithms consist of efficient time-to-depth conversion algorithms, based on Dijkstra-like Fast Marching Methods, as well as level set and ray tracing algorithms. The relation between the approaches and the algorithms in 2D and 3D are outlined in Fig. 1.2 (a) and (b) respectively.
4. Finally, we test our algorithms in 2D and 3D on a collection of examples.

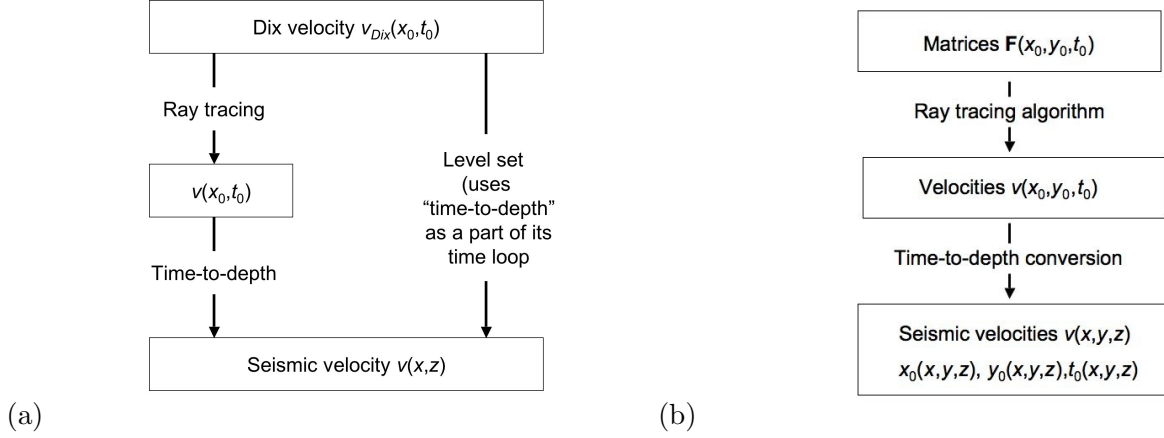


Figure 1.2. The approaches and the algorithms: (a) 2D, (b) 3D.

1.2 Background: settings and terminology

1.2.1 Seismic images

Seismic images are the images of the interior of the earth. They are crucial for the oil and gas industry. Obtaining seismic images of the subsurface structures where the layers are non-horizontal and the speed of propagation of the sound waves changes laterally is a very challenging and important problem: oil and gas tends to accumulate in such places. In particular, oil traps often occur around salt bodies. Salt bodies may have complicated shapes, and the sound speed inside them is typically high in comparison with that in the surrounding rock.

Seismic data, or the input data for seismic images, are collected as follows (see J. Claerbout [2000]). The basic equipment for reflection seismic prospecting is a source for impulsive sound waves, a receiver, and a multichannel waveform display system. A survey line is defined along the earth surface. For example, one might have a ship pulling a receiver cable behind it and activating the source every 25 meters. The waves sent by the source split at every medium discontinuity they encounter into reflected and deflected waves. The

first discontinuity the waves reach is the sea bed. The receivers record the amplitudes of the upcoming reflected waves. These records are the seismic data.

In geophysics, there are two commonly used coordinate systems: one is called time coordinates (or time migration coordinates), and the second is called depth coordinates (see Fig. 1.3, left and right respectively). Let A be a subsurface point. Suppose the point A explodes at time $t = 0$. Let the sound wave first reach the surface at a point x_0 at time $t = t_0$. Then (x_0, t_0) are its time coordinates. Its depth coordinates (x, z) are the location x of its projection onto the surface and its depth z .

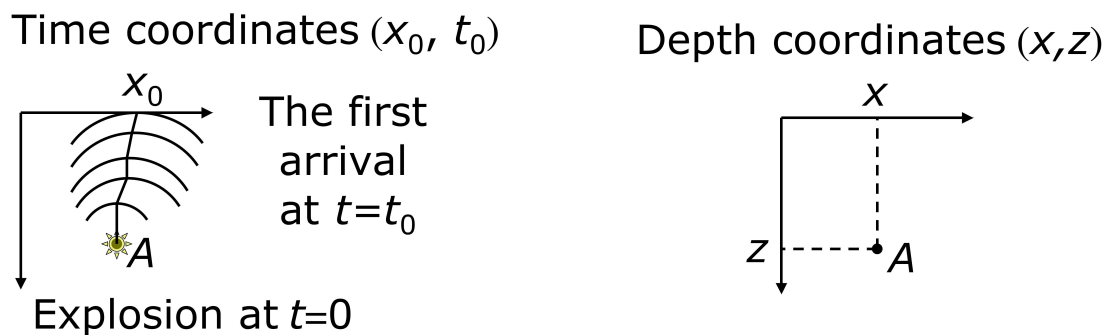


Figure 1.3. The time coordinates (left), and the depth coordinates (right).

Traditionally, seismic imaging is called migration. This name comes from the pre-computer era, when seismic images were obtained in a very simple and cheap way. Those who performed seismic imaging observed that the dipping (having a slope) reflectors on the recording plane moved to different locations (migrated) and changed their slope on the image plane.

There are two types of seismic imaging: time migration and depth migration. Time migration is fast and efficient but:

- it is adequate only for places where the seismic velocity depends primarily on the

depth, while oil tends to lie, and all interesting phenomena tend to occur, in the areas where the flat horizontal structures inside the earth are distorted;

- it produces images in the time coordinates which relate to the depth coordinates in a very subtle way if the velocity varies laterally.

Depth migration produces images in the depth coordinates and it is adequate for arbitrary places. But one needs to know the seismic velocity to implement it. Naturally, the seismic velocity is never known. Typically, it is found by "guessing and trying". The pros and cons of these two types of seismic imaging are summarized in Table 1.1.

Table 1.1. Comparison of time migration and depth migration

	Time migration	Depth migration
<i>Adequate for</i>	mild lateral velocity variation	arbitrary lateral velocity variation
<i>Implementation requires</i>	seismic data	seismic data and seismic velocity
<i>Produces images in</i>	time coordinates	depth coordinates

An example of a time migrated image¹ is shown in Fig. 1.4. This is an image of a section from the Gulf of Mexico. Note that it is in the time coordinates. The black and white stripes (the color depends of the phase of the waves at which they reach the receivers) represent the reflectors - the layers inside the earth. The gray area at the top indicates an absence of reflectors there. We can conclude that this area corresponds to the water of the Gulf of Mexico. The messy area at the bottom right corner indicates a failure to image correctly the reflectors located there. This is probably due to the fact that the signals from those reflectors first reach the surface beyond the area equipped with receivers.

Seismic imaging is based on the Huygens principle (see e.g. Yilmaz [2001]). Each point of a reflector is considered as a secondary source. The recorded seismic data are a superposition of the spherical waves coming from each reflecting point. Hence, if one were able to extract the responses coming from each reflecting point from the set of the recorded data, sum them with appropriate weights, and put the results into the appropriate point in

¹Courtesy of Dr. S. Fomel.

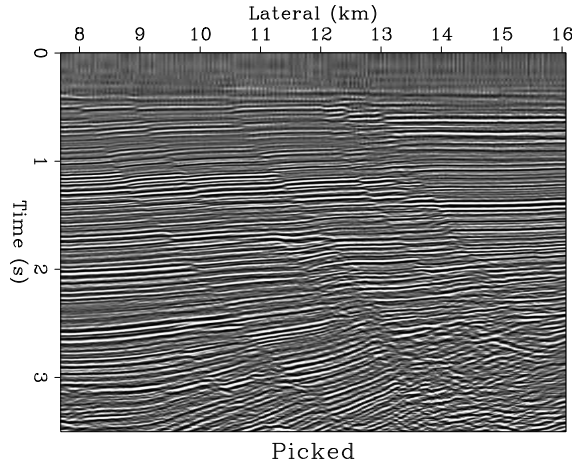


Figure 1.4. An example of a time migrated image.

the image, one would get the image of the earth. Naturally, in order to be able to find the response shapes from each reflecting point, one needs to have a guess about the propagation speed of the waves - the seismic velocities.

Time migration is based on a strong assumption about the responses in the recording plane coming from each point of the reflectors: they are assumed to have the same shape as they do in the case of constant velocity. The width of these shapes is defined by the time migration velocities, which are routinely determined in the process of time migration. On the one hand, this assumption allows one to obtain seismic images without a priori knowledge of the seismic velocity. On the other hand, it makes this type of imaging accurate only for mild lateral velocity variation.

Depth migration makes no assumption about the response shapes. As a result, on the one hand, one needs to know a priori the seismic velocities. On the other hand, once a velocity model is built, one can accurately image the earth with arbitrarily complex structures inside it and severe lateral velocity variation.

1.2.2 High frequency approximation

The seismic ray theory (see e.g. Červený [2001]; Popov [2002]), underlying this thesis, is based on the high frequency approximation of the wave equation. In this section, we will explain the connection between the Eikonal equation and the high frequency approximation to the wave equation, following the discussion in (Popov [2002]).

Consider the wave equation

$$\Delta P(x, y, z; t) = \frac{1}{v^2} \frac{\partial^2}{\partial t^2} P(x, y, z; t). \quad (1.2)$$

First suppose that the velocity v is constant. Let us seek a solution in the form of a plane wave

$$P = Ae^{i\phi}, \quad i^2 = -1, \quad \phi = -\omega t + k_1 x + k_2 y + k_3 z, \quad (1.3)$$

where A , ω , k_1 , k_2 , k_3 are constants. The plane wave given by equation (1.3) with a nonzero amplitude A satisfies the wave equation (1.2) if and only if the following dispersion relation holds

$$\frac{\omega^2}{v^2} = k_1^2 + k_2^2 + k_3^2. \quad (1.4)$$

Plane wave solutions play a remarkable role in mathematical physics because many types of solutions can be presented as a superposition of plane waves. Obviously, a plane wave solution does not exist if the velocity varies. But suppose that the velocity varies slowly. In this case, it is natural to seek a solution for the wave equation in a form of the so-called deformed plane wave

$$P = A(x, y, z) e^{i\phi(x, y, z, t)}, \quad (1.5)$$

where the amplitude A is no longer constant but depends on coordinates, and ϕ is no longer a linear function. Furthermore, we will consider a wave field harmonic in time, i.e.,

$$P(x, y, z, t) = e^{-i\omega t} U(x, y, z). \quad (1.6)$$

Inserting equation (1.6) into wave equation (1.2) we get for U the Helmholtz equation

$$\left(\Delta + \frac{\omega^2}{v^2} \right) U = 0. \quad (1.7)$$

Let us derive the Eikonal and transport equations. Now $v = v(x, y, z)$ and we seek a solution of the form

$$U = e^{i\omega T(x,y,z)} A(x, y, z), \quad (1.8)$$

where T is called eikonal, A is the amplitude, and the circular frequency ω is supposed to be a large parameter. Substituting equation (1.8) into the Helmholtz equation (1.7) we obtain

$$\left(\Delta + \frac{\omega^2}{v^2}\right) U = e^{i\omega T} \left(\omega^2 \left(\frac{1}{v^2} - (\nabla T)^2 \right) A + i\omega(2\nabla T \cdot \nabla A + A\Delta T) + \Delta A \right). \quad (1.9)$$

We suppose that ω is a large parameter of the problem and impose the following equations

$$(\nabla T)^2 = \frac{1}{v^2} \quad \text{(Eikonal equation);} \quad (1.10)$$

$$2\nabla T \cdot \nabla A + A\Delta T = 0 \quad \text{(Transport equation),} \quad (1.11)$$

in order to eliminate larger terms. But ΔA remains and we have no chance to eliminate it. Thus, in this case, we are not able to satisfy the equation exactly!

Thus, if the circular frequency ω is large, the solution of the form

$$P(x, y, z, t) = A(x, y, z) e^{-i\omega t} e^{i\omega T(x,y,z)}, \quad (1.12)$$

where the eikonal $T(x, y, z)$ is a solution of Eikonal equation (1.10) and the amplitude $A(x, y, z)$ is a solution of transport equation (1.11), is an approximate solution of the wave equation (1.2). This approximation is called the *high frequency approximation*.

1.3 Time migration coordinates and image rays

For many decades, imaging of the earth interior was based on the assumption that the velocity inside the earth depends only on the depth and that the subsurface structures are horizontal or, at worst, planar with the same dipping angle. To obtain more complex structure distortions, Hubral Hubral [1977] introduced the concept of the *image ray*, which gives the connection between the time migration coordinates and the regular Cartesian coordinates.

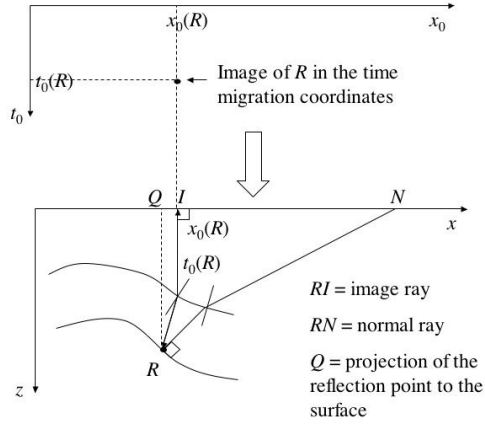


Figure 1.5. Image rays and time migration coordinates.

To explain this idea, we begin with the high frequency approximation applied to the wave equation (1.1), in which the wave front $T(x, y, z)$ propagates according to the Eikonal equation (see e.g. Popov [2002]):

$$|\nabla T(x, y, z)|^2 = \frac{1}{v^2(x, y, z)}. \quad (1.13)$$

The characteristics of the Eikonal equation can be viewed as rays. Among all rays starting at a subsurface point R and reaching earth's surface (Fig. 1.5), some have minimal travel time. These rays are called image rays, and it is easy to see that they must arrive perpendicular to the surface. The ray RI in Fig. 1.5 is one such image ray. Thus, we may characterize the point R in one of two coordinate systems: either (1) its natural Cartesian coordinates \mathbf{x} or (2) the point on the surface such that an image ray leaving \mathbf{x}_0 and traveling for a given time reaches the point R . The former given are called depth coordinates, while the latter are called time migration coordinates.

The conventional time migration coordinates are (\mathbf{x}_0, t_0) where \mathbf{x}_0 is the escape location of the image ray, and t_0 is the doubled (two-way) travel time along it. Note that the lateral position of the point R in the time migrated image is determined namely by the escape location of the image ray I rather than by its projection Q to the surface (Fig. 1.5).

1.4 Travel time approximation

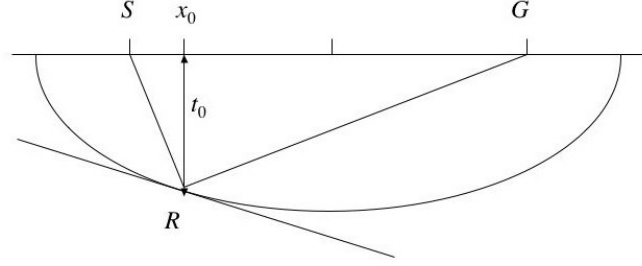


Figure 1.6. Travel time approximation.

Let S be a source and G be a receiver (Fig. 1.6), and let R be the reflection point. In the simplest case, we assume that the velocity v inside the earth is constant. Then the total travel time from S to R and from R to G is:

$$t_{SR} + t_{RG} = \sqrt{\frac{t_0^2}{4} + \frac{|\mathbf{x}_0 - S|^2}{v^2}} + \sqrt{\frac{t_0^2}{4} + \frac{|\mathbf{x}_0 - G|^2}{v^2}}. \quad (1.14)$$

The ellipse in Fig. 1.6 is the locus of the reflection points A such that the total travel time $t_{SA} + t_{AG}$ is the same as $t_{SR} + t_{RG}$.

In the general case where the velocity inside the earth is arbitrary, formula (1.14) serves as a starting point for a travel time approximation for time migration (namely its modern variant called "prestack time migration") (see Yilmaz [2001]). In this approximation, \mathbf{x}_0 is the escape location of an image ray from the reflection point R , and t_0 is the two-way travel time along it. The velocity v present in formula (1.14) is replaced with parameters with dimensions of velocity that depend on \mathbf{x}_0 and t_0 . These parameters are called the *migration velocities* and denoted by $v_m(\mathbf{x}_0, t_0)$. They are chosen to provide the best fit to formula (1.14) in the process of time migration. Thus, formula (1.14) suggests the approximation

$$t(S, G, \mathbf{x}_0, t_0) = \sqrt{\frac{t_0^2}{4} + \frac{|\mathbf{x}_0 - S|^2}{v_m^2(\mathbf{x}_0, t_0)}} + \sqrt{\frac{t_0^2}{4} + \frac{|\mathbf{x}_0 - G|^2}{v_m^2(\mathbf{x}_0, t_0)}}. \quad (1.15)$$

In the case where the velocity inside the earth depends only on the depth and the distance

between the source and the receiver is small, the migration velocity $v_m(\mathbf{x}_0, t_0)$ is the root-mean-square (RMS) velocity (Dix [1955]), given by

$$v_m(t_0) = \sqrt{\frac{1}{t_0} \int_0^{t_0} v^2(z(\tau)) d\tau}. \quad (1.16)$$

1.5 Emerging wave front

In this section, our aim is to justify the travel time approximation given by formula (1.15).

1.5.1 General 3D case

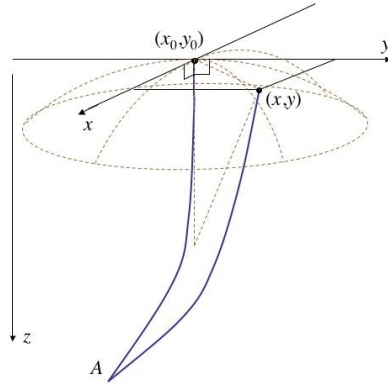


Figure 1.7. Emerging wave front.

Consider an emerging wave front from a point source A (Fig. 1.7) (see Hubral and Krey [1980]). Let the image ray arrive at the surface point (x_0, y_0) at time t_0 (here t_0 is the one-way travel time along the image ray). The travel time from A to the surface along some other ray close to the image ray, arriving at the surface point (x, y) , is given by the Taylor expansion

$$t(x, y) = t_0 + \frac{1}{2} \Delta \mathbf{x}^T \mathbf{\Gamma} \Delta \mathbf{x} + O(\delta^3), \quad (1.17)$$

where $\Delta \mathbf{x} = \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix}$, $\mathbf{\Gamma}$ is the matrix of the second derivatives of $t(x, y)$ evaluated at the point (x_0, y_0) and $\delta = \sqrt{(x - x_0)^2 + (y - y_0)^2}$. From geometrical considerations, one can obtain (see Hubral and Krey [1980]) a relation between the matrix $\mathbf{\Gamma}$ and the matrix \mathbf{R} of the radii of curvature of the emerging wave front, namely

$$\mathbf{\Gamma}^{-1} = \mathbf{R}v(x_0, y_0), \quad (1.18)$$

where $v(x_0, y_0) = v(x = x_0, y = y_0, z = 0)$ is the velocity at the surface point (x_0, y_0) . For convenience, we will work with the inverse of the matrix $\mathbf{\Gamma}$ which we denote by \mathbf{K} :

$$\mathbf{K} = \mathbf{\Gamma}^{-1} = \mathbf{R}v(x_0, y_0). \quad (1.19)$$

1.5.2 2D simplification

In the case where sources and receivers are arranged along some straight line, seismic imaging becomes a 2D problem, and equation (1.17) can be simplified to

$$t(x) = t_0 + \frac{1}{2}(x - x_0)^2 t_{xx}(x = x_0) + O(\delta^3) = t_0 + \frac{(x - x_0)^2}{2Rv(x_0)} + O(\delta^3), \quad (1.20)$$

where $v(x_0) \equiv v(x = x_0, z = 0)$. By squaring both sides of equation (1.20) we get:

$$t^2(x) = t_0^2 + (x - x_0)^2 t_0 t_{xx}(x = x_0) + O(\delta^3) = t_0^2 + (x - x_0)^2 \frac{t_0}{Rv(x_0)} + O(\delta^3). \quad (1.21)$$

Suppose we want to compute the total travel time from a source S to the reflection point A and from A to a receiver G . Using equation (1.21) we obtain:

$$t(x_0, t_0, S, G) = t_{SA} + t_{AG} = \sqrt{t_0^2 + (S - x_0)^2 \frac{t_0}{Rv(x_0)}} + \sqrt{t_0^2 + (G - x_0)^2 \frac{t_0}{Rv(x_0)}} + O(\delta^3). \quad (1.22)$$

Comparing equations (1.22) and (1.15) we see that the travel time approximation given by formula (1.15) follows from the Taylor expansion in 2D. Moreover, the migration velocity and the radius of curvature of the emerging wave front are converted through the relation

$$t_0 v_m^2(x_0, t_0) = v(x_0) R(x_0, t_0). \quad (1.23)$$

On the other hand, in 3D the travel time approximation given by formula (1.15) is not as straightforward. Instead, one can easily derive the following travel time formula from equations (1.17) and (1.18):

$$t(\mathbf{x}_0, t_0, S, G) = \sqrt{t_0^2 + t_0(S - \mathbf{x}_0)^T \mathbf{K}(\mathbf{x}_0, t_0)^{-1}(S - \mathbf{x}_0)} + \sqrt{t_0^2 + t_0(G - \mathbf{x}_0)^T \mathbf{K}(\mathbf{x}_0, t_0)^{-1}(G - \mathbf{x}_0)}. \quad (1.24)$$

However note that if the velocity depends only on the depth, the matrix \mathbf{K} is a multiple of the identity matrix, and hence formula (1.15) is the consequence of the Taylor expansion.

1.6 Dix inversion

Dix (Dix [1955]) established the first connection between the migration velocities and the seismic velocities for the case where the velocity depends only on the depth. He showed that the migration velocities are the root-mean-square (RMS) velocities if the distances between the sources and the receivers are small and developed the following inversion method. Consider an earth model as in Fig. 1.8. Let the layers be flat and horizontal, and the ve-

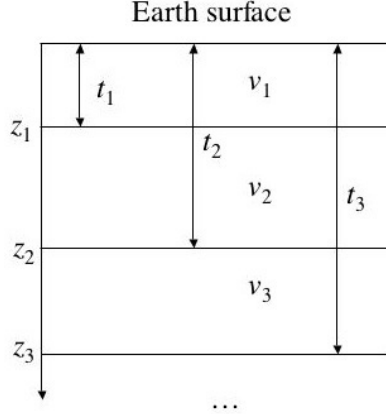


Figure 1.8. Dix inversion.

locity be constant within each layer. We are given the RMS velocities V_i and the travel times t_i , $i = 1, 2, \dots, n$, where V_i is the RMS velocity of the first i layers with respect to the

time, and t_i is the two-way vertical travel time from the earth surface to the bottom of the i -th layer. Then the layer velocities (typically called "interval velocities") v_i can be found successively from $i = 2$ to n :

$$v_i = \sqrt{\frac{V_i^2 t_i - V_{i-1}^2 t_{i-1}}{t_i - t_{i-1}}}. \quad (1.25)$$

The depths of the lower boundaries of the layers are:

$$z_i = z_{i-1} + v_i \frac{t_i - t_{i-1}}{2}. \quad (1.26)$$

Although it is derived for the cases in which the velocities are horizontally constant, in practice this Dix inversion is sometimes applied to find the interval velocities from the migration velocities in the case where the velocity varies laterally. For such cases, for the continuously changing velocity in 2D the Dix velocities are given by:

$$v(x_0, t_0) = \sqrt{\frac{\partial}{\partial t_0} (t_0 v_m^2(x_0, t_0))}. \quad (1.27)$$

Chapter 2

Forward modeling of time migration velocities

In this section we derive our main theoretical result: the relation between the migration velocities and the true seismic velocities in 2D and between the matrix \mathbf{K} in formula (1.17) in 3D. The result in 2D was established by mutual efforts of Sergey Fomel and the author, and then was extended to 3D by the author. We will also establish the stability of both the forward and the backward construction problem.

2.1 Paraxial ray tracing

For any ray propagating in a 3D medium with a smooth velocity (Fig. 2.1), we can call this ray *central* and attach a coordinate system (t, q_1, q_2) around it (see Popov [2002], Červený [2001]). Let t be the travel time along the central ray. For each moment of time t , we can draw a plane perpendicular to the central ray at the point which it reaches at time t . Pick two mutually orthogonal directions in this plane and call them \vec{e}_1 and \vec{e}_2 . Then the location of any point M in the space can be expressed as

$$\vec{r}_M = \vec{r}_0(t) + q_1\vec{e}_1 + q_2\vec{e}_2$$

for some t , q_1 and q_2 , where $\vec{r}_0(t)$ gives the point reached by the central ray at time t . If M is close enough to the ray, its location can be described by (t, q_1, q_2) uniquely.

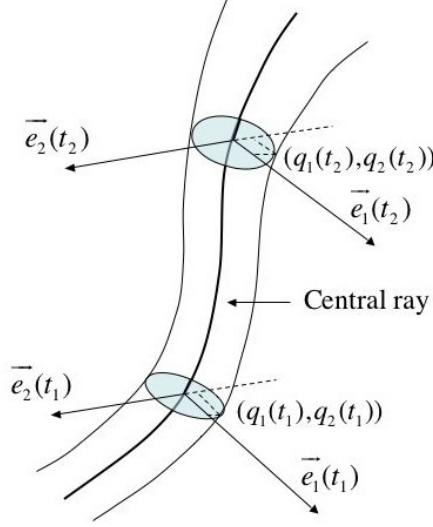


Figure 2.1. Paraxial ray tracing.

Suppose that the central ray is surrounded by a family of close rays and we want to write equations of those rays in terms of $q_1(t)$ and $q_2(t)$ (Fig. 2.1). In order to apply the Hamiltonian formalism, we need to introduce the generalized momenta p_1 and p_2 corresponding to the generalized coordinates q_1 and q_2 . We first note the fact that the central ray is a ray itself, and this imposes the following requirements on the evolution of \vec{e}_1 and \vec{e}_2 :

$$\frac{d\vec{e}_1}{dt} = \frac{\partial v(t, q_1, q_2)}{\partial q_1} \Big|_{q_1=q_2=0} \vec{\tau}, \quad \frac{d\vec{e}_2}{dt} = \frac{\partial v(t, q_1, q_2)}{\partial q_2} \Big|_{q_1=q_2=0} \vec{\tau},$$

where $\vec{\tau}$ is the unit tangent vector to the central ray (Popov [2002]). The ray equations in the Hamiltonian form are (Popov [2002], Popov and Pšenčík [1978], Červený [2001]):

$$\frac{d}{dt} \begin{pmatrix} q \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{0} & v_0^2 I_2 \\ -\frac{1}{v_0} \mathbf{V} & \mathbf{0} \end{pmatrix} \begin{pmatrix} q \\ p \end{pmatrix}. \quad (2.1)$$

Here v_0 is the velocity along the central ray, I_2 is the 2×2 identity matrix, and \mathbf{V} is a 2×2

matrix of the second derivatives of the velocity:

$$\mathbf{V}_{ij} = \frac{\partial^2 v(t, q_1, q_2)}{\partial q_i \partial q_j}, \quad i, j = 1, 2.$$

Suppose that the family of rays depends upon two parameters (α_1, α_2) . There are two important cases (see Fig. 2.2):

- All rays start perpendicular to the same plane. Then (α_1, α_2) can be chosen to be the initial coordinates (x_0, y_0) of the rays at this plane. We will call such a family of rays *telescopic*.
- All rays start at the same point, but in different directions. Then (α_1, α_2) can be chosen to be the initial momenta $(p_1(0), p_2(0))$ of the rays. We will call such a family the *point source family*.

Consider the following 2×2 matrices (Popov [2002], Červený [2001]):

$$\mathbf{Q}_{ij} \equiv \frac{\partial q_i}{\partial \alpha_j}, \quad \mathbf{P}_{ij} \equiv \frac{\partial p_i}{\partial \alpha_j}, \quad i, j = 1, 2. \quad (2.2)$$

The equations of time evolution for \mathbf{Q} and \mathbf{P} are the equations in variations for equation (2.1):

$$\frac{d}{dt} \begin{pmatrix} \mathbf{Q} \\ \mathbf{P} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & v_0^2 I_2 \\ -\frac{1}{v_0} \mathbf{V} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{Q} \\ \mathbf{P} \end{pmatrix}. \quad (2.3)$$

The initial conditions for the telescopic family of rays are

$$\mathbf{Q}(0) = I_2, \quad \mathbf{P}(0) = \mathbf{0}, \quad (2.4)$$

and for the point source family they are

$$\mathbf{Q}(0) = \mathbf{0}, \quad \mathbf{P}(0) = \frac{1}{v_0(0)} I_2, \quad (2.5)$$

where $v_0(0)$ is the velocity at the source point. The absolute value of the determinant of the matrix \mathbf{Q} has a nice geometrical sense (Popov [2002]):

$|\det \mathbf{Q}|$ is the geometrical spreading of the family of rays.

Let the central ray arrive orthogonal to some plane at a point (x_0, y_0) . Consider the matrix $\mathbf{\Gamma}$ of the second derivatives of the travel times of the family of rays around the central ray, evaluated at the point (x_0, y_0) . E.g., the central ray can be the image ray arriving to the earth surface. Then the matrix $\mathbf{\Gamma}$ is defined by formula (1.17) for the source point family of rays from the source point A as in Fig. 1.7. In Popov [2002], Červený [2001] it was shown that

$$\mathbf{\Gamma} = \mathbf{P}\mathbf{Q}^{-1} \quad (2.6)$$

and

$$\frac{d}{dt}\mathbf{\Gamma} = -v_0^2\mathbf{\Gamma}^2 - \frac{1}{v_0}\mathbf{V}. \quad (2.7)$$

For convenience, in the present work we will deal with the matrix $\mathbf{K} = \mathbf{\Gamma}^{-1}$, which is the matrix of radii of curvature of the wave front scaled by the velocity at the image ray, namely

$$\mathbf{K} = v_0\mathbf{R} = \mathbf{Q}\mathbf{P}^{-1}. \quad (2.8)$$

One can easily derive from equation (2.7) that the time evolution of \mathbf{K} is given by:

$$\frac{d}{dt}\mathbf{K} = v_0^2\mathbf{I}_2 + \frac{1}{v_0}\mathbf{K}\mathbf{V}\mathbf{K}. \quad (2.9)$$

For the point source family of rays the initial conditions for the matrix \mathbf{K} are:

$$\mathbf{K}(0) = \mathbf{0}. \quad (2.10)$$

2.2 Relation between the matrix \mathbf{K} and the true seismic velocities in 3D

We have established the relation between the matrices \mathbf{K} and the seismic velocities in 3D formulated in Theorem 1 below. The matrix \mathbf{K} in formula (1.24) is a matrix of parameters depending on \mathbf{x}_0 and t_0 , which can be estimated from the measurements. Theorem 1 provides a connection between the matrix \mathbf{K} and the true seismic velocity at the subsurface point \mathbf{x} reached by the image ray arriving at \mathbf{x}_0 and traced backwards for time t_0 (Fig. 2.2).

Theorem 1 *Let an image ray starting from a subsurface point \mathbf{x} (Fig. 2.2) arrive at the earth surface point \mathbf{x}_0 at time t_0 . Designate this ray to be central. Let the matrix $\mathbf{K}(\mathbf{x}_0, t_0)$ be evaluated at the surface for a point source family of rays around the image ray, starting at the same point \mathbf{x} . Suppose there is also a telescopic family of rays around the image ray starting perpendicular to the earth surface which we trace backwards w.r.t. the image ray for time t_0 and compute the matrices \mathbf{Q} and \mathbf{P} . Let $\mathbf{Q}(\mathbf{x}_0, t_0)$ be the matrix \mathbf{Q} for the telescopic family of rays evaluated at the time t_0 (i.e., at the subsurface point \mathbf{x}) in this backward tracing. Then*

$$\frac{\partial}{\partial t_0} \mathbf{K}(\mathbf{x}_0, t_0) = v^2(\mathbf{x}(\mathbf{x}_0, t_0)) (\mathbf{Q}(\mathbf{x}_0, t_0)^T \mathbf{Q}(\mathbf{x}_0, t_0))^{-1}. \quad (2.11)$$

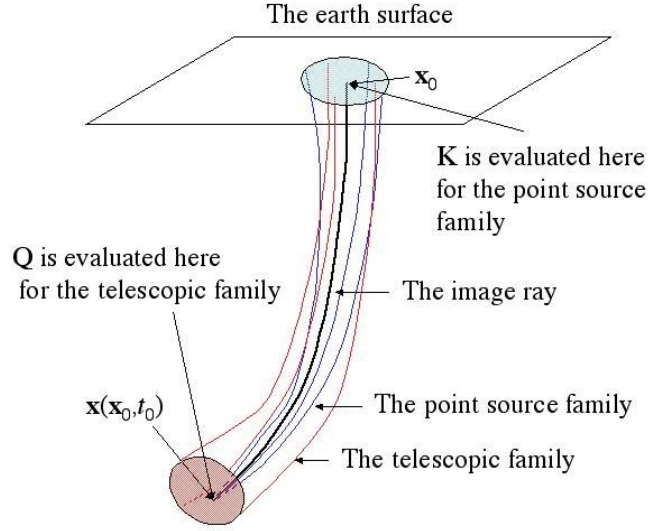


Figure 2.2. Illustration for Theorem 1.

Proof Let an image ray arrive at the surface point \mathbf{x}_0 at time t_1 . Fix a moment of time $t_0 < t_1$ and consider a point source family of rays starting at the subsurface point $\mathbf{x}(\mathbf{x}_0, t_0)$

which the image ray passes at time t_0 . Introduce the following notations:

$$\mathbf{X} = \begin{pmatrix} \mathbf{Q} \\ \mathbf{P} \end{pmatrix} = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \\ P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}, \quad \mathbf{A}(t) = \begin{pmatrix} \mathbf{0} & v_0^2 I_2 \\ -\frac{1}{v_0} \mathbf{V} & \mathbf{0} \end{pmatrix}.$$

Let \mathbf{X}_* be the 4×4 matrix of derivatives of \mathbf{X} with respect to the initial conditions:

$$\mathbf{X}(t_0) = \begin{pmatrix} Q_{110} & Q_{120} \\ Q_{210} & Q_{220} \\ P_{110} & P_{120} \\ P_{210} & P_{220} \end{pmatrix} :$$

$$\mathbf{X}_* = \begin{pmatrix} \frac{\partial Q_{11}}{\partial Q_{110}} & \frac{\partial Q_{11}}{\partial Q_{210}} & \frac{\partial Q_{12}}{\partial Q_{120}} & \frac{\partial Q_{12}}{\partial Q_{220}} \\ \frac{\partial Q_{21}}{\partial Q_{110}} & \frac{\partial Q_{21}}{\partial Q_{210}} & \frac{\partial Q_{22}}{\partial Q_{120}} & \frac{\partial Q_{22}}{\partial Q_{220}} \\ \frac{\partial P_{11}}{\partial P_{110}} & \frac{\partial P_{11}}{\partial P_{210}} & \frac{\partial P_{12}}{\partial P_{120}} & \frac{\partial P_{12}}{\partial P_{220}} \\ \frac{\partial P_{21}}{\partial P_{110}} & \frac{\partial P_{21}}{\partial P_{210}} & \frac{\partial P_{22}}{\partial P_{120}} & \frac{\partial P_{22}}{\partial P_{220}} \end{pmatrix}.$$

Note that since each of the columns of \mathbf{X} is a linear independent solution of equation (2.1) the derivatives not included into \mathbf{X}_* are zeros. $\mathbf{X}(t)$ and $\mathbf{X}_*(t)$ are solutions of the following initial value problems:

$$\frac{d\mathbf{X}}{dt} = \mathbf{A}(t)\mathbf{X}, \quad \mathbf{X}(t_0) = \frac{1}{v(t_0)} \begin{pmatrix} \mathbf{0} \\ I_2 \end{pmatrix}, \quad (2.12)$$

where $v(t_0) = v(\mathbf{x}(\mathbf{x}_0, t_0))$, and

$$\frac{d\mathbf{X}_*}{dt} = \mathbf{A}(t)\mathbf{X}_*, \quad \mathbf{X}_*(t_0) = I_4. \quad (2.13)$$

Denote the solution of equation (2.13) by $\mathbf{B}(t_0; t_1)$ as it is done in Červený [2001]:

$$\mathbf{B}(t_0; t_1) = \begin{pmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \\ \mathbf{P}_1 & \mathbf{P}_2 \end{pmatrix},$$

where $\mathbf{Q}_i, \mathbf{P}_i, i = 1, 2$ are 2×2 matrices. $\begin{pmatrix} \mathbf{Q}_1 \\ \mathbf{P}_1 \end{pmatrix}$ satisfies the initial conditions corresponding to a telescopic point, and $\begin{pmatrix} \mathbf{Q}_2 \\ \mathbf{P}_2 \end{pmatrix}$ satisfies the initial conditions corresponding

to a normalized point source. $\mathbf{B}(t_0, t_1)$ is called *the propagator matrix*. Then the solution of (2.12) is:

$$\mathbf{X}(t) = \frac{1}{v(t_0)} \begin{pmatrix} \mathbf{Q}_2 \\ \mathbf{P}_2 \end{pmatrix}. \quad (2.14)$$

Now turn to the matrix \mathbf{K} : $\mathbf{K}(t_0; t_1) = \mathbf{Q}(t_0; t_1)\mathbf{P}(t_0; t_1)^{-1} = \mathbf{Q}_2\mathbf{P}_2^{-1}$.

Shift the initial time t_0 by $-\Delta t$. Then, according to equation (2.12) at time t_0

$$\begin{aligned} \mathbf{Q}(t_0 - \Delta t; t_0) &= 0 + \Delta t v^2(t_0) \frac{1}{v(t_0)} I_2 + O((\Delta t)^2), \\ \mathbf{P}(t_0 - \Delta t; t_0) &= \frac{1}{v(t_0)} I_2 + O((\Delta t)^2). \end{aligned}$$

Hence the change in the initial conditions for equation (2.12) is:

$$\Delta \mathbf{Q}_0 = v_0 \Delta t I_2 + O((\Delta t)^2), \quad \Delta \mathbf{P}_0 = \mathbf{0} + O((\Delta t)^2). \quad (2.15)$$

Then

$$\begin{aligned} \mathbf{K}(t_0 - \Delta t; t_1) &= \mathbf{K}(t_0; t_1) + \sum_{i,j=1}^2 \frac{\partial \mathbf{K}}{\partial Q_{ij0}} \Delta Q_{ij0} + \sum_{i,j=1}^2 \frac{\partial \mathbf{K}}{\partial P_{ij0}} \Delta P_{ij0} + O((\Delta t)^2) \\ &= \mathbf{K}(t_0; t_1) + \left(\frac{\partial \mathbf{K}}{\partial Q_{110}} + \frac{\partial \mathbf{K}}{\partial Q_{220}} \right) v(t_0) \Delta t + O((\Delta t)^2). \end{aligned} \quad (2.16)$$

Let us find the partial derivatives in the expression above:

$$\frac{\partial \mathbf{K}}{\partial Q_{ii0}} = \frac{\partial \mathbf{Q}}{\partial Q_{ii0}} \mathbf{P}^{-1} - \mathbf{Q} \mathbf{P}^{-1} \frac{\partial \mathbf{P}}{\partial Q_{ii0}} \mathbf{P}^{-1}, \quad i = 1, 2. \quad (2.17)$$

In terms of the entries of the matrix $\mathbf{B}(t_0; t_1)$

$$\frac{\partial \mathbf{K}}{\partial Q_{110}} + \frac{\partial \mathbf{K}}{\partial Q_{220}} = v_0 (\mathbf{Q}_1 \mathbf{P}_2^{-1} - \mathbf{Q}_2 \mathbf{P}_2^{-1} \mathbf{P}_1 \mathbf{P}_2^{-1}). \quad (2.18)$$

In Červený [2001] the symplectic property of the matrix $\mathbf{B}(t_0; t_1)$ was proved:

$$\mathbf{B}^T \mathbf{J} \mathbf{B} = \mathbf{J}, \quad (2.19)$$

where \mathbf{J} is the 4×4 matrix

$$\mathbf{J} = \begin{pmatrix} \mathbf{0} & I_2 \\ -I_2 & \mathbf{0} \end{pmatrix}.$$

To simplify formula (2.18) we will use the following consequences of the symplectic property (2.19):

$$\mathbf{P}_2^T \mathbf{Q}_1 - \mathbf{Q}_2^T \mathbf{P}_1 = I_2, \quad \mathbf{P}_2^T \mathbf{Q}_2 = \mathbf{Q}_2^T \mathbf{P}_2. \quad (2.20)$$

Then the matrix expression in equation (2.18) simplifies to:

$$\begin{aligned} & \mathbf{Q}_1 \mathbf{P}_2^{-1} - \mathbf{Q}_2 \mathbf{P}_2^{-1} \mathbf{P}_1 \mathbf{P}_2^{-1} = \\ & (\mathbf{P}_2^T)^{-1} \mathbf{P}_2^T \mathbf{Q}_1 \mathbf{P}_2^{-1} - (\mathbf{P}_2^T)^{-1} \mathbf{P}_2^T \mathbf{Q}_2 \mathbf{P}_2^{-1} \mathbf{P}_1 \mathbf{P}_2^{-1} = \\ & (\mathbf{P}_2^T)^{-1} (\mathbf{P}_2^T \mathbf{Q}_1 - \mathbf{P}_2^T \mathbf{Q}_2 \mathbf{P}_2^{-1} \mathbf{P}_1) \mathbf{P}_2^{-1} = \\ & (\mathbf{P}_2^T)^{-1} (\mathbf{P}_2^T \mathbf{Q}_1 - \mathbf{Q}_2^T \mathbf{P}_2 \mathbf{P}_2^{-1} \mathbf{P}_1) \mathbf{P}_2^{-1} = \\ & (\mathbf{P}_2^T)^{-1} (\mathbf{P}_2^T \mathbf{Q}_1 - \mathbf{Q}_2^T \mathbf{P}_1) \mathbf{P}_2^{-1} = \\ & (\mathbf{P}_2^T)^{-1} \mathbf{P}_2^{-1}. \end{aligned} \quad (2.21)$$

Substituting Eqn. (2.21) to Eqn. (2.18) and then to Eqn. (2.16) we get:

$$\mathbf{K}(t_0 - \Delta t; t_1) = \mathbf{K}(t_0; t_1) + \Delta t v^2(t_0) (\mathbf{P}_2^T)^{-1} \mathbf{P}_2^{-1} + O((\Delta t)^2). \quad (2.22)$$

Then the derivative of \mathbf{K} with respect to the initial time is:

$$-\frac{\partial \mathbf{K}(t_0; t_1)}{\partial t_0} = v^2(t_0) (\mathbf{P}_2^T)^{-1} \mathbf{P}_2^{-1}. \quad (2.23)$$

In Červený [2001] the following reciprocity property was proved:

$$\mathbf{P}_2^T(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{Q}_1(\mathbf{x}_2, \mathbf{x}_1), \quad (2.24)$$

where $\mathbf{x}_1, \mathbf{x}_2$ are the end points of the central ray. Applying it to equation (2.23) and taking the time reverse into account we obtain formula (2.11).

2.3 Relation between the Dix velocities and the true seismic velocities in 2D

In 2D the matrices \mathbf{Q} , \mathbf{P} and \mathbf{K} become scalars which we denote by Q , P and K respectively. K is the radius of curvature of the wave front scaled by the velocity at the

central ray: $K = vR$. The time evolution of Q , P and K are given by:

$$\frac{d}{dt} \begin{pmatrix} Q \\ P \end{pmatrix} = \begin{pmatrix} 0 & v_0^2 \\ -\frac{v_{qq}}{v_0} & 0 \end{pmatrix} \begin{pmatrix} Q \\ P \end{pmatrix}, \quad \frac{dK}{dt} = v^2 + \frac{v_{qq}}{v} K^2. \quad (2.25)$$

In a similar way as it was done in 3D, it can be proven that

$$\frac{\partial}{\partial t_0} K(x_0, t_0) = \frac{v^2(x(x_0, t_0), z(x_0, t_0))}{Q^2(x_0, t_0)}. \quad (2.26)$$

Then taking into account the definition of the Dix velocity (1.27) and the relation (1.23) between the migration velocities and the radius of curvature of the emerging wave front we have the following:

Theorem 2 *Let an image ray arrive to the earth surface point x_0 at time t_0 from a subsurface point (x, z) . Suppose there is a telescopic family of rays around the image ray starting perpendicular to the earth surface which we trace backwards w.r.t. the image ray for time t_0 and compute the quantities Q and P . Let $Q(x_0, t_0)$ be the quantity Q for the telescopic family of rays evaluated at the time t_0 (i.e., at the subsurface point (x, z)) in this backward tracing. Then the Dix velocity $v_{Dix}(x_0, t_0)$ is the ratio of the true seismic velocity $v(x, z)$ and the absolute value of $Q(x_0, t_0)$:*

$$v_{Dix}(x_0, t_0) = \frac{v(x(x_0, t_0), z(x_0, t_0))}{|Q(x_0, t_0)|}. \quad (2.27)$$

Note that here, t_0 is the one-way travel time along the image ray and that we denote the depth direction by z .

2.4 Analytical example

Take a velocity field with a constant gradient of slowness squared¹:

$$\frac{1}{v^2(x, z)} = s^2(x, z) = s_0^2 + 2gx. \quad (2.28)$$

In such a medium, an analytic result can be produced by ray tracing. We will demonstrate that

¹We thank Dr. S. Fomel for this example.

$$(i) \quad \frac{dQ}{dt} = v^2 P, \quad \frac{dP}{dt} = -\frac{v_{qq}}{v} Q;$$

$$(ii) \quad K = Q/P;$$

(iii) Theorem 2 holds, i.e.,

$$\frac{\partial K}{\partial t_0} = \frac{1}{P^2} = \frac{v^2}{Q^2},$$

where P relates to the point source family traced toward the surface, and Q relates to the telescopic family traced from the surface.

We begin by noting that the rays are parabolic trajectories (there is an analogy with particles moving in constant gravitational field):

$$\begin{aligned} H &= \frac{1}{2}|\vec{p}|^2 - \frac{1}{2}s^2(x, z), \\ \frac{dx}{d\sigma} &= p_x, \quad \frac{dz}{d\sigma} = p_z, \\ \frac{dp_x}{d\sigma} &= g, \quad \frac{dp_z}{d\sigma} = 0. \end{aligned} \tag{2.29}$$

Hence,

$$x(\sigma) = \frac{g\sigma^2}{2} + A\sigma + B, \quad z(\sigma) = C\sigma + D, \tag{2.30}$$

where $(x')^2 + (z')^2 = p_x^2 + p_z^2 = s_0^2 + 2gx$. Consider a ray \overrightarrow{MO} arriving normally to the surface ($z = 0$) at the origin at $\sigma = \sigma_0$ (Fig 2.3). It is easy to see that this ray is given by

$$x(\sigma) = \frac{g}{2}(\sigma_0 - \sigma)^2, \tag{2.31}$$

$$z(\sigma) = s_0(\sigma_0 - \sigma).. \tag{2.32}$$

The slowness along this ray is

$$p_x^2 + p_z^2 = (x'(\sigma))^2 + (z'(\sigma))^2 = g^2(\sigma_0 - \sigma)^2 + s_0^2 \equiv s(\sigma). \tag{2.33}$$

(i) Let us compute Q and P for a point source family of rays at the surface by their definitions, i.e., from the geometrical considerations. The ray \overrightarrow{MO} is an image ray. Designate it to be central and consider a narrow point source family of rays starting at $(x = \frac{g\sigma_0^2}{2}, z = s\sigma_0)$. Pick one ray, the ray \overrightarrow{MA} (Fig. 2.3) from this family, and find its equations. Let the initial momentum of the ray \overrightarrow{MA} differs from the one of the ray \overrightarrow{MO} by a small angle α . We will

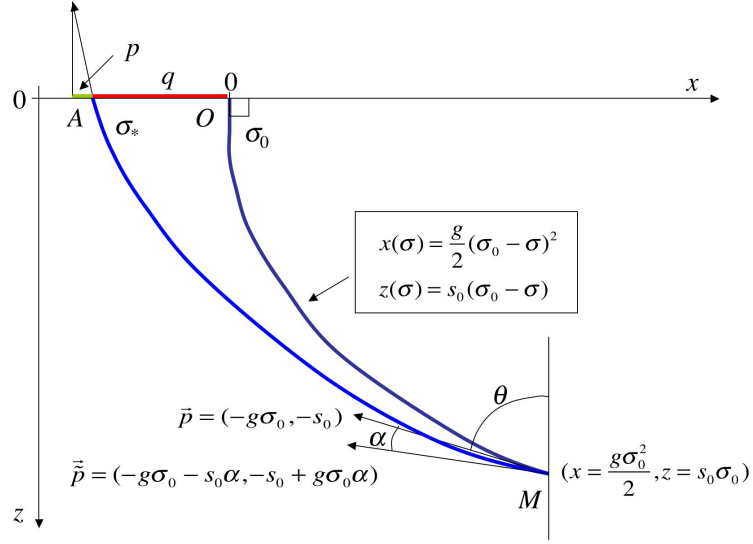


Figure 2.3. Illustration for the analytical example.

denote by letters with tilde ($\tilde{}$) the quantities related to the ray \overrightarrow{MA} . At the initial point, the absolute value of momentum

$$|\vec{\tilde{p}}(0)| = \sqrt{s_0^2 + g^2\sigma_0^2} \equiv s(0). \quad (2.34)$$

Hence,

$$\begin{aligned} \tilde{p}_x &= s(0) \sin(\theta + \alpha) = s(0)(\sin \theta + \alpha \cos \theta + O(\alpha^2)) \\ &= s(0)\left(-\frac{g\sigma_0}{s(0)} - \alpha \frac{s_0}{s(0)} + \dots\right) \\ &= -g\sigma_0 - \alpha s_0 + O(\alpha^2), \end{aligned} \quad (2.35)$$

$$\begin{aligned} \tilde{p}_z &= s(0) \cos(\theta + \alpha) = s(0)(\cos \theta - \alpha \sin \theta + O(\alpha^2)) \\ &= s(0)\left(-\frac{s_0}{s(0)} + \alpha \frac{g\sigma_0}{s(0)} + \dots\right) \\ &= -s_0 + \alpha g\sigma_0 + O(\alpha^2). \end{aligned} \quad (2.36)$$

Using equations 2.30, 2.34 and 2.35 we obtain

$$\tilde{x}(\sigma) = \frac{g}{2}(\sigma_0 - \sigma)^2 - \alpha s_0 \sigma + O(\alpha^2), \quad (2.37)$$

$$\tilde{z}(\sigma) = s_0(\sigma_0 - \sigma) + \alpha g \sigma_0 \sigma + O(\alpha^2). \quad (2.38)$$

The ray \overrightarrow{MA} hits the surface at $\sigma = \sigma_*$ which can be found from the following equation:

$$\tilde{z}(\sigma_*) = s_0(\sigma_0 - \sigma_*) + \alpha g \sigma_0 \sigma_* + O(\alpha^2) = 0.$$

Hence,

$$\sigma_* = \frac{s_0 \sigma_0}{s_0 - g \sigma_0 \alpha} + O(\alpha^2) = \sigma_0 + \frac{g \sigma_0^2}{s_0} \alpha + O(\alpha^2). \quad (2.39)$$

Plugging equation (2.39) into equation (2.37) we find the location where the ray \overrightarrow{MA} hits the surface:

$$\tilde{x}(\sigma_*) = -s_0 \sigma_0 \alpha + O(\alpha^2). \quad (2.40)$$

Choosing the direction of the axis q in the ray coordinates opposite to the direction of the x -axis, we find

$$q = s_0 \sigma_0 \alpha.$$

Now we can find Q at the point O by its definition:

$$Q \equiv \left. \frac{dq}{d\alpha} \right|_{\alpha=0} = s_0 \sigma_0. \quad (2.41)$$

The momentum p corresponding to the ray coordinate q is the projection of the momentum of a ray onto the axis q which is orthogonal to the momentum of the central ray. At the surface, the central ray \overrightarrow{MO} has a vertical momentum. Hence, for the ray \overrightarrow{MA} , $p = -p_x(\sigma_*)$:

$$p = -\tilde{p}_x(\sigma_*) = -x'(\sigma_*) = g(\sigma_0 - \sigma_*) + \alpha s_0 + O(\alpha^2) = \alpha \frac{s_0^2 - g^2 \sigma_0^2}{s_0} + O(\alpha^2). \quad (2.42)$$

By definition, P at the point O is given by

$$P \equiv \left. \frac{dp}{d\alpha} \right|_{\alpha=0} = \frac{s_0^2 - g^2 \sigma_0^2}{s_0}. \quad (2.43)$$

Now let us check that Q and P satisfy equations (2.25) (Claim (iii)). Draw a line normal to the ray \overrightarrow{MO} at the location $(x(\sigma) = \frac{g}{2}(\sigma_0 - \sigma)^2, z(\sigma) = s_0(\sigma_0 - \sigma))$ (Fig. 2.5). This line

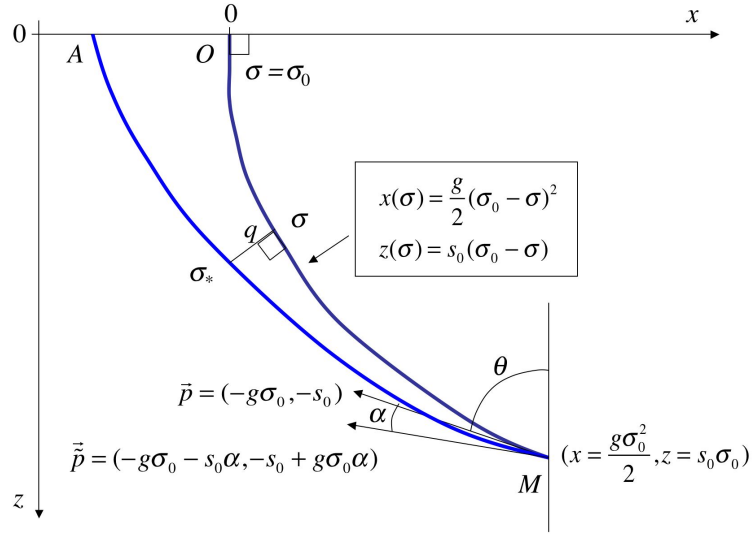


Figure 2.4. Illustration for the analytical example.

is given by

$$\begin{aligned} x(q) &= \frac{g}{2}(\sigma_0 - \sigma)^2 - \frac{s_0}{s(\sigma)}q, \\ z(q) &= s_0(\sigma_0 - \sigma) + \frac{g(\sigma_0 - \sigma)}{s(\sigma)}q, \end{aligned} \quad (2.44)$$

where q is the signed distance from the central ray, the q in the ray coordinates in the paraxial ray tracing. Find the distance between the rays \overrightarrow{MA} and \overrightarrow{MO} along this line:

$$\tilde{x}(\sigma_*) = \frac{g}{2}(\sigma_0 - \sigma_*)^2 - \alpha s_0 \sigma_* + O(\alpha^2) = \frac{g}{2}(\sigma_0 - \sigma)^2 - \frac{s_0}{s(\sigma)}q, \quad (2.45)$$

$$\tilde{z}(\sigma_*) = s_0(\sigma_0 - \sigma_*) + \alpha g \sigma_0 \sigma_* + O(\alpha^2) = s_0(\sigma_0 - \sigma) + \frac{g(\sigma_0 - \sigma)}{s(\sigma)}q. \quad (2.46)$$

We find σ_* from equation (2.46):

$$\sigma_* = \frac{-s\sigma + \frac{g(\sigma_0 - \sigma)}{s(\sigma)}q}{-s_0 + \alpha g \sigma_0} + O(\alpha^2) = \sigma + \frac{q \sigma \sigma_0}{s_0} \alpha - \frac{g(\sigma_0 - \sigma)}{s(\sigma) s_0} q + O(\alpha^2) + O(\alpha q). \quad (2.47)$$

Substitute it into equation (2.45):

$$\begin{aligned}
\frac{g}{2}(\sigma_0 - \sigma)^2 - \frac{s_0}{s(\sigma)}q &= \frac{g}{2}\left(\sigma_0 - \sigma - \frac{q\sigma\sigma_0}{s_0}\alpha + \frac{g(\sigma_0 - \sigma)}{s(\sigma)s_0}q\right)^2 - \alpha s_0\sigma + O(\alpha^2) + O(\alpha q), \\
-\frac{s_0}{s(\sigma)}q &= -\frac{g^2(\sigma_0 - \sigma)\sigma\sigma_0}{s_0}\alpha + \frac{g^2(\sigma_0 - \sigma)^2}{s(\sigma)s_0}q + O(\alpha^2) + O(\alpha q) + O(q^2), \\
q\left(\frac{s_0}{s(\sigma)} + \frac{g^2(\sigma_0 - \sigma)^2}{s(\sigma)s_0} + O(\alpha)\right) &= \alpha\left(s_0\sigma + \frac{g^2(\sigma_0 - \sigma)\sigma\sigma_0}{s_0}\right) + O(\alpha^2) + O(q^2), \\
q\frac{s(\sigma)}{s_0} &= \alpha\frac{s_0^2 + g^2(\sigma_0 - \sigma)\sigma_0}{s_0}\sigma + O(\alpha^2) + O(q^2).
\end{aligned}$$

Hence,

$$q = \alpha\frac{s_0^2 + g^2(\sigma_0 - \sigma)\sigma_0}{s(\sigma)}\sigma + O(\alpha^2). \quad (2.48)$$

Thus,

$$Q \equiv \frac{dq}{d\alpha}\Big|_{\alpha=0} = \frac{s_0^2 + g^2(\sigma_0 - \sigma)\sigma_0}{s(\sigma)}\sigma. \quad (2.49)$$

According to equation (2.25),

$$\frac{dQ}{dt} = v^2 P.$$

Using the fact that $\frac{dt}{d\sigma} = s^2$ and equation (2.49) we compute

$$P = \frac{s_0^4 + g^4\sigma(\sigma_0 - \sigma)^3 + s_0^2g^2\sigma_0(2\sigma_0 - 3\sigma)}{s^3(\sigma)}. \quad (2.50)$$

At the surface we have

$$Q(\sigma_0) = s_0\sigma_0, \quad (2.51)$$

$$P(\sigma_0) = \frac{s_0^2 - g^2\sigma_0^2}{s_0}. \quad (2.52)$$

We see that they coincide with the ones computed by the definitions (equations (2.41) and (2.43)).

The velocity given by equation (2.28) can be rewritten in terms of the ray coordinates $\{\sigma, q\}$ as follows:

$$\frac{1}{v^2(\sigma, q)} = s^2(\sigma, q) = s_0^2 + 2g(x(\sigma) - q \cos \theta(\sigma)) = s_0^2 + 2gx(\sigma) + 2g\frac{s_0q}{s(\sigma)} = s^2(\sigma) + 2gq\frac{s_0}{s(\sigma)}. \quad (2.53)$$

Differentiating $v(\sigma, q)$ twice w.r.t. q we get:

$$v_{qq} = \frac{3g^2s_0^2}{s^7(\sigma)}. \quad (2.54)$$

Using equation (2.50) we compute:

$$\begin{aligned}
\frac{dP}{dt} &= \frac{1}{s^2(\sigma)} \frac{dP}{d\sigma} \\
&= -\frac{3g^4\sigma_0(\sigma_0 - \sigma)^2}{s^5(\sigma)} - \frac{3s_0^2g^2\sigma_0}{s^5(\sigma)} + 3g^2(\sigma_0 - \sigma) \frac{s_0^4 + g^4\sigma_0(\sigma_0 - \sigma)^3 + s_0^2g^2\sigma_0(2\sigma_0 - 3\sigma)}{s^7(\sigma)} \\
&= \frac{-3g^4s_0^2\sigma_0(\sigma_0 - \sigma)^2 - 3g^6\sigma_0(\sigma_0 - \sigma)^4 - 3s_0^4g^2\sigma_0 - 3s_0^2g^4\sigma_0(\sigma_0 - \sigma)^2}{s^7(\sigma)} \\
&\quad + \frac{3s_0^4g^2(\sigma_0 - \sigma) + 3g^6\sigma_0(\sigma_0 - \sigma)^4 + 6s_0^2g^4\sigma_0^2(\sigma_0 - \sigma) - 9s_0^2g^4\sigma_0\sigma(\sigma_0 - \sigma)}{s^7(\sigma)} \\
&= \frac{-6g^4s_0^2\sigma_0(\sigma_0 - \sigma)^2 - 3s_0^4g^2\sigma + 6s_0^2g^4\sigma_0^2(\sigma_0 - \sigma) - 9s_0^2g^4\sigma_0\sigma(\sigma_0 - \sigma)}{s^7(\sigma)} \\
&= \frac{g^4s_0^2\sigma_0(-6\sigma_0^2 + 12\sigma_0\sigma - 6\sigma^2 + 6\sigma_0^2 - 6\sigma_0\sigma - 9\sigma_0\sigma + 9\sigma^2) - 3s_0^4g^2\sigma}{s^7(\sigma)} \\
&= \frac{g^4s_0^2\sigma_0(3\sigma^2 - 3\sigma_0\sigma) - 3s_0^4g^2\sigma}{s^7(\sigma)} \\
&= -\frac{3g^2s_0^2(g^2\sigma_0(\sigma_0 - \sigma) + s_0^2)}{s^7(\sigma)} \tag{2.55}
\end{aligned}$$

Comparing equations (2.55) and (2.49) we see that

$$\frac{dP}{dt} = -\frac{3g^2s_0^2}{s^7}Qs_0 = -\frac{v_{qq}}{v}Q.$$

Thus, we have demonstrated that Q and P satisfy equation (2.25) (Claim (i)).

(ii) Now we will show that $K = Q/P$ (Claim (ii)). Let us first compute K at the surface from its definition: $K = vR$ where R is the radius of curvature of the emerging wave front:

$$\begin{aligned}
K = vR &= \frac{1}{s_0} \lim_{\alpha \rightarrow 0} \frac{q}{\arg(\vec{p}(\sigma_*), \vec{\tilde{p}}(\sigma_*))} = \frac{1}{s_0} s_0\sigma_0\alpha \lim_{\alpha \rightarrow 0} \left| \frac{\tilde{p}_z(\sigma_*)}{\tilde{p}_x(\sigma_*)} \right| \\
&= \sigma_0\alpha \lim_{\alpha \rightarrow 0} \frac{s_0 - \alpha g\sigma_0^2}{\alpha \frac{s_0^2 - g^2\sigma_0^2}{s_0}} \frac{s_0^2\sigma_0}{s_0^2 - g^2\sigma_0^2}. \tag{2.56}
\end{aligned}$$

Computing the ratio of Q and P at the surface we find

$$K(\sigma_0) = \frac{Q(\sigma_0)}{P(\sigma_0)} = \frac{s_0^2\sigma_0}{s_0^2 - g^2\sigma_0^2} \tag{2.57}$$

which coincides with the result given by equation (2.56).

(iii) Check equation (2.23).

$$t_0 = \int_0^{\sigma_0} ((s_0^2 + g^2(\sigma_0 - \sigma)^2) d\sigma. \tag{2.58}$$

Hence,

$$\frac{dt_0}{d\sigma_0} = s_0^2 + \int_0^{\sigma_0} 2g^2(\sigma_0 - \sigma)d\sigma = s_0^2 + .g^2\sigma_0^2 \equiv s^2(0). \quad (2.59)$$

Therefore,

$$\begin{aligned} \frac{\partial K}{\partial t_0} &= \frac{\partial K}{\partial \sigma_0} \frac{1}{s^2(0)} = \left(\frac{s_0^2}{s_0^2 - g^2\sigma_0^2} + \frac{2g^2s_0^2\sigma_0^2}{(s_0^2 - g^2\sigma_0^2)^2} \right) \frac{1}{s^2(0)} \\ &= \frac{s_0^2s^2(0)}{(s_0^2 - g^2\sigma_0^2)^2} \frac{1}{s^2(0)} = \frac{s_0^2}{(s_0^2 - g^2\sigma_0^2)^2} \equiv \frac{1}{P^2}. \end{aligned} \quad (2.60)$$

We did not get minus here due to the fact that we made σ_0 a varying parameter, while the initial σ was fixed to be 0 rather than the other way around as in the proof of Theorem 1.

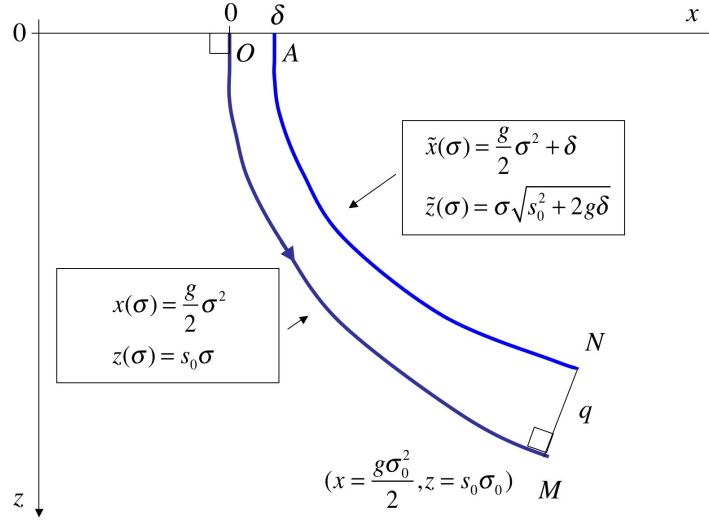


Figure 2.5. Illustration for the analytical example.

Now let us show that equation (2.26) holds, i.e. Theorem 2. Trace the ray \overrightarrow{MO} backward till $\sigma = \sigma_0$. Then the ray \overrightarrow{OM} is given by

$$x(\sigma) = \frac{g}{2}\sigma^2, \quad (2.61)$$

$$z(\sigma) = s_0\sigma. \quad (2.62)$$

Along with this ray, consider a telescopic family of ray starting normal to the surface. Pick a ray \overrightarrow{AN} from this family. Let this ray starts at the location $x_0 = \delta$ (Fig. 2.5). One can

easily find that the ray \overrightarrow{AN} is given by

$$\tilde{x}(\sigma) = \frac{g}{2}\sigma^2 + \delta, \quad (2.63)$$

$$\tilde{z}(\sigma) = \sigma\sqrt{s_0^2 + 2g\delta}. \quad (2.64)$$

Compute Q for the ray \overrightarrow{OM} at the point $(x = \frac{g}{2}\sigma_0^2, z = s_0\sigma_0)$. The equations of a line passing through this point and orthogonal to the ray \overrightarrow{OM} are

$$x(q) = \frac{g\sigma_0^2}{2} - \frac{s_0}{s(\sigma_0)}q, \quad (2.65)$$

$$z(q) = s_0\sigma_0 + \frac{g\sigma_0}{s(\sigma_0)}q,$$

where $s(\sigma_0) = \sqrt{s_0^2 + g^2\sigma_0^2}$, and q is the signed distance from the ray \overrightarrow{OM} along this line.

Find q corresponding to the intersection of this line with the ray \overrightarrow{AN} :

$$\frac{g\sigma_0^2}{2} - \frac{s_0}{s(\sigma_0)}q = \frac{g}{2}\sigma_*^2 + \delta, \quad (2.66)$$

$$s_0\sigma_0 + \frac{g\sigma_0}{s(\sigma_0)}q = \sigma_*\sqrt{s_0^2 + 2g\delta}. \quad (2.67)$$

We express σ_* from equation (2.67) and plug it into equation (2.66):

$$\frac{g}{2} \left(\frac{s_0^2\sigma_0^2 + \frac{g^2\sigma_0^2}{s^2(\sigma_0)}q^2 + \frac{2s_0\sigma_0^2g}{s(\sigma_0)}q}{s_0^2 + 2g\delta} \right) + \delta = \frac{g\sigma_0^2}{2} - \frac{s_0}{s(\sigma_0)}q.$$

Hence,

$$q = \frac{\frac{g\sigma_0^2}{2} \left(\frac{s_0^2 + 2g\delta - s_0^2}{s_0^2 + 2g\delta} \right) - \delta}{\frac{s_0}{s(\sigma_0)} \frac{g^2\sigma_0^2 + s_0^2 + 2g\delta}{s_0^2 + 2g\delta}} + O(\delta^2) = \frac{g^2\sigma_0^2\delta - s_0^2\delta}{s_0^2 + g^2\sigma_0^2 + 2g\delta} \frac{s_0}{s(\sigma_0)} + O(\delta^2). \quad (2.68)$$

Then,

$$Q \equiv \frac{dq}{d\delta}_{\delta=0} = \frac{g^2\sigma_0^2 - s_0^2}{s^2(\sigma_0)} \frac{s(\sigma_0)}{s_0} = \frac{g^2\sigma_0^2 - s_0^2}{s(\sigma_0)s_0}. \quad (2.69)$$

Comparing equations (2.60) and (2.69) we get

$$\frac{\partial K}{\partial t_0} = \frac{s_0^2}{(s_0^2 - g^2\sigma_0^2)^2} \equiv \frac{v^2(M)}{Q^2}. \quad (2.70)$$

2.5 Statement of the inverse problem

We state an inverse problem both in 2D and in 3D. Here, t_0 will denote the one-way travel time along the image ray.

2.5.1 The inverse problem in 2D

Suppose there is an image ray arriving at each surface point x_0 , $x_{min} \leq x_0 \leq x_{max}$. For any $0 \leq t_0 \leq t_{max}$, trace the image ray backward for time t_0 together with a small telescopic family of rays. Let the image ray being traced backward reach a subsurface point (x, z) at time t_0 . Denote by $v(x_0, t_0)$ the velocity at the point (x, z) , and by $Q(x_0, t_0)$ the quantity Q for the corresponding telescopic family at the point (x, z) . We are given $v_{Dir}(x_0, t_0) = \frac{v(x(x_0, t_0), z(x_0, t_0))}{|Q(x_0, t_0)|} \equiv f(x_0, t_0)$, $x_{min} \leq x_0 \leq x_{max}$, $0 \leq t_0 \leq t_{max}$. We need to find $v(x, z)$, the velocity inside the domain covered with the image rays arriving to the surface in the interval $[x_{min}, x_{max}]$.

The first question is whether this problem is well-posed. In the next sections, we will show that both the direct problem (given $v(x, z)$ find $f(x_0, t_0)$) and the inverse problem (given $f(x_0, t_0)$ find $v(x, z)$) are ill-posed. We will use the notation $f(x_0, t_0) \equiv \frac{v(x(x_0, t_0), z(x_0, t_0))}{|Q(x_0, t_0)|}$ rather than $v_{Dir}(x_0, t_0)$ to emphasize that f is computed as the ratio $v/|Q|$ rather than from the optimal migration velocities.

Ill-posedness of the direct problem

Direct Problem: Given $v(x, z)$, $x_{min} \leq x \leq x_{max}$, $x_{min} < 0$, $x_{max} > 0$, $z \geq 0$ and t_{max} find $f(x_0, t_0) = \frac{v(x_0, t_0)}{|Q(x_0, t_0)|}$, $x_{min} \leq x_0 \leq x_{max}$, $0 \leq t_0 \leq t_{max}$.

We shall show that small changes in $v(x, z)$ can lead to large changes in $f(x_0, t_0)$. Take $v(x, z) = 1$ and $\tilde{v}(x, z) = 1 + a \cos(kx)$, $-1 \leq x \leq 1$. Then

$$\|\tilde{v} - v\|_\infty = a.$$

Obviously, $f(x_0, t_0) = 1$ for $v(x, z) = 1$. Compute $\tilde{f}(x_0, t_0)$ for $\tilde{v}(x, z)$ at $x_0 = 0$. As the image ray arriving at $x_0 = 0$ is straight, we have that

$$v_{qq} = v_{xx}(x = 0) = -ak^2.$$

Then we have:

$$\frac{dQ}{dt_0} = (1 + a^2)P, \quad \frac{dP}{dt_0} = \frac{ak^2}{1+a}Q, \quad Q(0) = 1, \quad P(0) = 0.$$

Therefore,

$$\frac{d^2Q}{dt_0^2} = \frac{ak^2(1+a^2)}{1+a}Q, \quad Q(0) = 1, \quad \frac{dQ}{dt_0} = 0.$$

Hence,

$$Q(t_0) = \cosh \omega t_0, \quad \omega = \sqrt{\frac{ak^2(1+a^2)}{1+a}}.$$

Pick $k = \frac{1}{a}$ and let a tend to zero. Then $\frac{1}{\sqrt{2a}} < \omega < \sqrt{\frac{2}{a}}$, and

$$f(x_0 = 0, t_0) = \frac{1}{\cosh \omega t_0} < \frac{1}{\cosh \frac{t_0}{\sqrt{2a}}}.$$

Hence

$$\|\tilde{f}(x_0, t_0) - f(x_0, t_0)\|_\infty > 1 - \frac{1}{\cosh \frac{t_{max}}{\sqrt{2a}}} > \frac{1}{2}$$

for a small enough. Thus, we have shown that arbitrarily small changes in the velocity $v(x, y)$ may lead to significant changes in $f(x_0, t_0)$, i.e., *the direct problem is physically unstable in the max norm.*

Ill-posedness of the inverse problem

Inverse Problem: Given $f(x_0, t_0) = \frac{v(x_0, t_0)}{|Q(x_0, t_0)|}$, $x_{min} \leq x_0 \leq x_{max}$, $0 \leq t_0 \leq t_{max}$, find $v(x, z)$, the velocity inside the domain covered with the image rays arriving to the surface in the interval $[x_{min}, x_{max}]$.

Here we shall prove that the corresponding discrete problem is ill-posed: Given $f(x_{0i}, t_k)$, $i = 0, 1, \dots, n-1$, $k = 0, 1, \dots, p-1$, $x_{0i} = x_{min} + i\Delta x$, $t_k = k\Delta t$, where $\Delta x = (x_{max} - x_{min})/(n-1)$, $\Delta t = t_{max}/(p-1)$ respectively, find $v(x_i, z_j)$, $i = 0, 1, \dots, n-1$, $j = 0, 1, \dots, m-1$.

Let $x_{min} = -L$ and $x_{max} = L$ and n be odd so that $x = 0$ is one of the grid lines. Suppose we are given the following two discrete arrays: (1) $f(x_{0i}, t_k) = 1$ and (2) $\tilde{f}(x_{0i}, t_k) = 1$ if $x_{0i} \neq 0$ and $\tilde{f}(x_{0i}, t_k) = b > 1$ if $x_{0i} = 0$. Then

$$\|f(x_{0i}, t_k) - \tilde{f}(x_{0i}, t_k)\|_\infty = b - 1. \quad (2.71)$$

For $f(x_{0i}, t_k) = 1$ $v(x, y) = 1$. Let us find a velocity $\tilde{v}(x, z)$ such that the exact values of \tilde{f} for it coincides with $\tilde{f}(x_{0i}, t_k)$ on the mesh. Let the mesh step in x_0 be Δx . We will look for $\tilde{v}(x, z)$ in the following form: pick $0 < \alpha \leq \Delta x$ and set $\tilde{v}(x, z) = 1$ if $|x| \geq \alpha$, and

$$\tilde{v}(x, z) = v(x, t_0(z)) = 1 + (\tilde{v}(0, t_0) - 1) \exp \left(1 - \frac{1}{1 - \left(\frac{x}{\alpha}\right)^2} \right)$$

if $|x| < \alpha$. Here $v(0, t_0)$ is to be found. Note that

$$\tilde{v}_{xx}(0, t_0) = -\frac{2}{\alpha^2}(\tilde{v}(0, t_0) - 1). \quad (2.72)$$

Since $\tilde{f}(0, t_k) = \frac{\tilde{v}(0, t_0)}{Q(0, t_0)} = b$,

$$Q(0, t_0) = \frac{\tilde{v}(0, t_0)}{b}. \quad (2.73)$$

Due to the symmetry of our $v(x, z)$, the ray starting at $x_0 = 0$ perpendicular to the surface is straight. Let us write the IVP for Q and P for this ray:

$$\begin{aligned} \frac{dQ}{dt_0} &= v^2 P, & Q(T=0) &= 1, \\ \frac{dP}{dt_0} &= -\frac{v_{xx}}{v} Q, & P(T=0) &= 0. \end{aligned}$$

Here $v(t_0) \equiv \tilde{v}(0, t_0)$. Taking into account relation (2.73) and using Eqn. (2.72) we get:

$$\begin{aligned} \frac{dv}{dt_0} &= bv^2 P, & v(t_0=0) &= b, \\ \frac{dP}{dt_0} &= \frac{2}{\alpha^2 b}(v-1), & P(t_0=0) &= 0 \end{aligned} \quad (2.74)$$

Along with IVP (2.74) consider the following IVP:

$$\begin{aligned} \frac{dw}{dt_0} &= bu, & w(t_0=0) &= b, \\ \frac{du}{dt_0} &= \frac{2}{\alpha^2 b}(w-1), & u(t_0=0) &= 0. \end{aligned} \quad (2.75)$$

Solving IVP (2.75) we find:

$$w(t_0) = 1 + (b-1) \cosh \left(\frac{t_0 \sqrt{2}}{\alpha} \right).$$

Then by a variant of a comparison theorem, on the interval $[0, T_*)$ where the solution to IVP (2.74) exists, $v(t_0) > w(t_0)$. Hence, $\tilde{v}(0, t_0)$ either blows up, or reaches its maximum at t_{max} . Hence we conclude that

$$\|v(x, z) - \tilde{v}(x, z)\|_\infty > (b-1) \cosh \left(\frac{t_{max} \sqrt{2}}{\alpha} \right). \quad (2.76)$$

Comparing formulae (2.71) and (2.76) we see that for any b we can pick $\alpha = \min\{\Delta x, \frac{(b-1)t_{max}\sqrt{2}}{3}\}$ and hence make the left-hand side of Eqn. (2.76) greater than 1. Thus we have shown that *the inverse problem is numerically unstable in the max norm.*

Eulerian formulation of the inverse problem

The inverse problem stated in Section 2.5 can be formulated in a different, Eulerian way. Consider the mapping between the Cartesian coordinates (x, z) and the time migration coordinates (x_0, t_0) . The functions $x_0(x, z)$ and $t_0(x, z)$ satisfy the following system of equations:

$$|\nabla x_0|^2 = \left(\frac{\partial x_0}{\partial x}\right)^2 + \left(\frac{\partial x_0}{\partial z}\right)^2 = \frac{1}{Q^2(x, z)}, \quad (2.77)$$

$$\nabla x_0 \cdot \nabla t_0 = \frac{\partial x_0}{\partial x} \frac{\partial t_0}{\partial x} + \frac{\partial x_0}{\partial z} \frac{\partial t_0}{\partial z} = 0, \quad (2.78)$$

$$|\nabla t_0|^2 = \left(\frac{\partial t_0}{\partial x}\right)^2 + \left(\frac{\partial t_0}{\partial z}\right)^2 = \frac{1}{v^2(x, z)}. \quad (2.79)$$

Equation (2.77) follows from the definition of Q . Equation (2.78) indicates that the curves $t_0=\text{const}$ are orthogonal to the image rays, and will be derived in Section 3.1.1 below. Equation (2.79) is the Eikonal equation.

The input data are

$$v_{Dix}^2(x_0, t_0) = \frac{v^2(x(x_0, t_0), z(x_0, t_0))}{Q^2(x(x_0, t_0), z(x_0, t_0))}. \quad (2.80)$$

The boundary conditions are:

$$x_0(x, 0) = x, \quad t_0(x, 0) = 0, \quad Q(x, 0) = 1, \quad v(x, 0) = v_{Dix}(x_0 = x, t_0 = 0). \quad (2.81)$$

2.5.2 The inverse problem in 3D

Suppose there is an image ray arriving at each surface point (x_0, y_0) , $x_{min} \leq x \leq x_{max}$, $y_{min} \leq y \leq y_{max}$. For any $0 \leq t_0 \leq t_{max}$, trace the image ray backward for time t_0 together with a small telescopic family of rays. Let the image ray reach a subsurface point (x, y, z) after being traced backward for time t_0 . Let $v(x, y, z)$ be the velocity at the point (x, y, z)

, and $Q(x_0, y_0, t_0)$ be the matrix Q for the small telescopic family at the point (x, y, z) . We are given

$$\frac{\partial K(x_0, y_0, t_0)}{\partial t_0} = v^2(x, y, z)(Q^T(x_0, y_0, t_0)Q(x_0, y_0, t_0))^{-1} \equiv F(x_0, y_0, t_0),$$

$x_{min} \leq x \leq x_{max}$, $y_{min} \leq y \leq y_{max}$, $0 \leq t_0 \leq t_{max}$. We need to find $v(x, y, z)$, the velocity inside the earth in the domain covered with the image rays arriving to the surface to the rectangle $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$.

Chapter 3

Numerical algorithms in 2D

In this section we propose three numerical algorithms. We will start with an efficient time-to-depth conversion algorithm. The input for it is $v(x_0, t_0) \equiv v(x(x_0, t_0), z(x_0, t_0))$. The output is $v(x, z)$, $x_0(x, z)$ and $t_0(x, z)$. This algorithm is an essential part of our other two algorithms which produce $v(x, z)$ from $v_{Dix}(x_0, t_0)$. The first of these two, based on a ray tracing approach, creates $v(x_0, t_0)$, the input for the time-to-depth algorithm. The second, based on a level set approach, uses it as a part of its time cycle. Also, if nothing else is available, Dix velocities can be used as the input for our time-to-depth conversion. The main advantage of our time-to-depth conversion algorithm is that it is very fast and robust.

3.1 Efficient time-to-depth conversion algorithm

In this section we will use notation T for t_0 to be consistent with the notations in the Eikonal equation (1.13). Also, we will deal with the reciprocal of the velocity $s(x, z)$ which we call *slowness* for convenience.

3.1.1 Eulerian formulation of the boundary value problem

Let (x, z) be a subsurface point (Fig. 3.1). Let $s(x, z)$ be the slowness at the point (x, z) . Let the image ray from (x, z) reach the surface at some point x_0 and let T be the one-way travel time from (x, z) to the surface point x_0 .

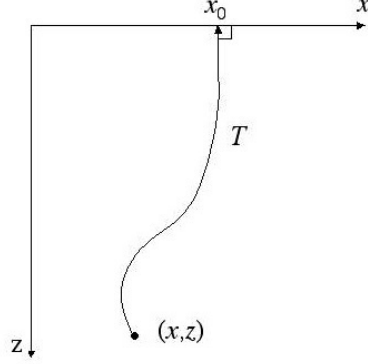


Figure 3.1. Section 3.1.1. Relation between (x, z) , x_0 and T .

Let $x_{min} \leq x_0 \leq x_{max}$, $0 \leq T \leq T_{max}$, $x_{min} \leq x \leq x_{max}$, $0 \leq z \leq z_{max}$. Given $s(x_0, T)$, our goal is to find $s(x, z)$, $x_0(x, z)$ and $T(x, z)$, i.e., the slowness at each subsurface point (x, z) , the escape location of the image ray from each subsurface point (x, z) , and the one-way travel time along each image ray. Thus, the input for this algorithm is given in the time domain (x_0, T) , and the desired output is in the depth domain (x, z) .

The functions $x_0(x, z)$ and $T(x, z)$ are well-defined in the case if the image rays do not intersect inside the domain in hand. If the image rays intersect, our algorithm will follow the first arrivals to the surface.

The functions $s(x_0, T)$, $x_0(x, z)$ and $T(x, z)$ are related according to the following system of PDE's:

$$|\nabla T|^2 = s^2(x_0, T) \equiv s(x_0(x, z), T(x, z)), \quad (3.1)$$

$$\nabla T \cdot \nabla x_0 = 0. \quad (3.2)$$

Equation (3.1) is the Eikonal equation with an unknown right-hand side. Equation (3.2) gives a connection between x_0 and T , and indicates that the curves $T=\text{const}$ are orthogonal to the image rays. We may derive this relation as follows:

We first note that the escape location x_0 is constant along each image ray. Hence the time derivative of x_0 along each image ray must be zero:

$$\frac{dx_0}{dT} = \frac{\partial x_0}{\partial x} \frac{dx}{dT} + \frac{\partial x_0}{\partial z} \frac{dz}{dT} = 0. \quad (3.3)$$

Writing the equations of the phase trajectories for the Hamiltonian

$$H = \frac{1}{2}|\nabla T|^2 - \frac{1}{2}s^2(x, z) = 0$$

given by the Eikonal equation, we have that

$$\frac{dx}{dT} = \frac{\partial T}{\partial x} \frac{1}{s^2}, \quad \frac{dz}{dT} = \frac{\partial T}{\partial z} \frac{1}{s^2}.$$

Substituting this into equation (3.3) we get:

$$\frac{\partial x_0}{\partial x} \frac{dx}{dT} + \frac{\partial x_0}{\partial z} \frac{dz}{dT} = \frac{1}{s^2} \nabla x_0 \cdot \nabla T = 0.$$

Hence, $\nabla x_0 \cdot \nabla T = 0$ as desired.

We also have boundary conditions for the system (3.2):

$$x_0(x, 0) = x, \quad T(x, 0) = 0, \quad s(x, 0) = s(x_0 = x, T = 0). \quad (3.4)$$

3.1.2 Numerical algorithm

The motivation and the main building block of this algorithm is Sethian's Fast Marching Method (Sethian [1996]) designed for solving a boundary value problem for the Eikonal equation with known right-hand-side. This method is a Dijkstra-type method, in that it

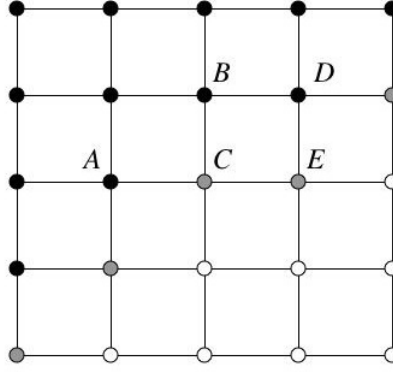


Figure 3.2. Fast Marching Method. Black, grey and white dots represent "Accepted", "Considered" and "Unknown" points respectively.

systematically advances the solution to the desired equation from known values to unknown values without iteration. Dijkstra's method, first developed in the context of computing a shortest path on a network, computes the solution in order $N \log N$, where N is the total number of points in the domain. The first extension of this approach to an Eikonal equation is due to Tsitsiklis (Tsitsiklis [1995]), who obtains a control-theoretic discretization of the Eikonal equation, which then leads to a causality relationship based on the optimality criterion. Tsitsiklis' algorithm evolved from studying isotropic min-time optimal trajectory problems, and involves solving a minimization problem to update the solution. A more recent, finite difference approach, based again on Dijkstra-like ordering and updating, was developed by Sethian (Sethian [1996, 1999A]) for solving the Eikonal equation. Sethian's Fast Marching Method evolved from studying isotropic front propagation problems, and involves an upwind finite difference formulation to update the solution. Both Tsitsiklis' method and the Fast Marching Method start with a particular (and different) coupled discretization and each shows that the resulting system can be decoupled through a causality property. In the particular case of a first order scheme on a square mesh, the resulting quadratic update equation at each grid point is the same for both methods. We refer the reader to these references for details on ordered upwind methods for Eikonal equations, as well as (Sethian and Vladimirsky [2003]) for a detailed discussion about the similarities and

differences between the two techniques. More recently, Sethian and Vladimirsky have built versions of a class of Ordered Upwind Methods, based on Dijkstra-like methodology, for solving the more general class of optimal control problems in which the speed/cost function depends on both position and direction, which leads to a convex Hamilton-Jacobi equation. See (Sethian and Vladimirsky [2003]) for details.

We now discuss the Fast Marching Method in more detail, since it will serve as a building block to our algorithm. In order to follow the physical propagation of information, an upwind scheme is used, and the solution is computed at points in order of increase of T . In order to achieve it, the points are divided into "Accepted", where T is computed and no longer can be updated and can be used for estimation of T at its neighbors; "Considered", where T is computed but may be updated in future and cannot be used for estimation of T at other points; "Unknown", where no value of T has been computed yet. At each time step a "Considered" point with the smallest value of T , determined by the heap sort, becomes "Accepted". Sethian used this approach to compute the solution of the Eikonal equation with known right-hand-side in a variety of settings including semiconductor processing, image segmentation, seismic wave propagation and robotic navigation: for details, see Sethian [1996, 1999A, 1999B].

In our case, the principal difference between previous work and our problem is that here, the right-hand side of the Eikonal equation is unknown. In terms of our numerical algorithm, we do not know the direction of propagation of information. This creates an issue which we will discuss in Section 3.1.3 below, however, we first outline the time-to-depth conversion algorithm.

Let us discretize and solve the system (3.1), (3.2) with boundary conditions (3.4). The input for the numerical algorithm is the matrix $s(x_{0i}, T_k)$, $i = 0, 1, \dots, n-1$, $k = 0, 1, \dots, p-1$. Denote the mesh steps in x_0 and T by hx and ΔT respectively. The mesh steps in x and z are hx and hz respectively. We define $s(x_0, T)$ beyond the mesh points by the bilinear interpolation. The output of the numerical algorithm are the matrices $s(x_i, z_j)$, $x_0(x_i, z_j)$ and $T(x_i, z_j)$, $i = 0, 1, \dots, n-1$, $j = 0, 1, \dots, m-1$. The algorithm is the following:

1. Mark the boundary (surface) points $(x_i = x_{0i}, z = 0)$ as "Accepted". Set $s(x_i, z_0 = 0) = s(x_0 = x_i, T = 0)$, $x_0(x, z = 0) = x_0$, $T(x, z = 0) = 0$ according to the boundary conditions. Mark the rest of the mesh points (x_i, z_j) as "Unknown".
2. Mark the "Unknown" points adjacent to the "Accepted" points as "Considered". We call two points adjacent (or nearest neighbors) if they are separated by one edge.
3. Compute or update tentative values of $s(x_i, z_j)$, $x_0(x_i, z_j)$ and $T(x_i, z_j)$ at the "Considered" points.

- (a) If a "Considered" point E has only one "Accepted" nearest neighbor D as in Fig. 3.2, then the values at E are found from the 1-point-update system:

$$\begin{aligned}
x_0(E) &= x_0(D), \\
T(E) - T(D) &= hs(x_0(D), T(E)), \\
s(E) &= s(x_0(E), T(E)), \\
T(E) &> T(D).
\end{aligned}
\tag{3.5}$$

Here H is either hx or hz depending on the arrangement of E and D .

- (b) If a "Considered" point has only two "Accepted" nearest neighbors and they are located so that it lies linearly between them, then we compute triplets of tentative values of s , x_0 and T for each of the two "Accepted" points and the "Considered" point from system (3.5), and then choose the triplet with the smallest value of T .
- (c) If a "Considered" point C has only two "Accepted" neighbors A and B not lying on the same grid line, as in Fig. 3.2, then the tentative values at C are found

from the 2-point-update system:

$$\begin{aligned}
& \frac{(T(C) - T(A))^2}{hx^2} + \frac{(T(C) - T(B))^2}{hz^2} = s^2(x_0(C), T(C)), \\
& \frac{(T(C) - T(A))(x_0(C) - x_0(A))}{hx^2} + \frac{(T(C) - T(B))(x_0(C) - x_0(B))}{hz^2} = 0, \\
& s(C) = s(x_0(C), T(C)), \\
& x_0(A) \leq x_0(C) \leq x_0(B), \\
& T(C) \geq \max\{T(A), T(B)\}.
\end{aligned} \tag{3.6}$$

We solve the first two equations in system (3.6) using a Newton solver.

- (d) If a "Considered" point has three or more "Accepted" nearest neighbors then we compute a triplet of tentative values for each possible couple of "Accepted" points forming a right triangle together with the "Considered" point such that the "Considered" point lies at its right angle, and choose the triplet with the smallest value of T .

4. Find a "Considered" point with the smallest tentative value of " T " and mark it as "Accepted". We use a heap sort to keep track of the tentative T values.
5. If the set of "Considered" points is not empty, return to 2.

3.1.3 Causality

At $T = 0$ the wave front is a segment of the straight line from $(x_0, 0)$ to $(x_{n-1}, 0)$. In order to propagate it correctly, we must compute the points in order of increase of T as given in Sethian's Fast Marching Method.

In the above our update principle, the 1-point update (3.5) *artificially puts point E on the image ray passing through D (Fig. 3.2) prescribing $x_0(E) = x_0(D)$, while the 2-point-update looks for the correct image ray (the correct value of x_0).*

At the moment when some "Unknown" point becomes "Considered", it has only one "Accepted" nearest neighbor. Therefore the tentative values at it are found from the 1-

point update system (3.5). Then, if it does not become “Accepted” by that time, it gets two “Accepted” neighbors lying on different grid lines. Then the values at it are found from the 2-point-update system (3.6). We emphasize that *we design our algorithm so that the 2-point-update values replace the 1-point-update values whenever it is possible independently of whether the new tentative value of T is smaller or larger*. Note that in the Fast Marching Method, the 2-point-update value never exceeds the 1-point-update value due to the fact that the slowness is known at each point. In our formulae (3.5) and (3.6) for 1- and 2-point-update respectively the slowness s in the right-hand side depends on T . Because of this, we cannot eliminate the situation where the value of T given by 1-point-update is smaller than the one given by 2-point-update. Such a situation is dangerous because the 1-point-update’s setting $x_0(E) = x_0(D)$ is correct only if the true velocity (slowness) at E is larger (smaller) than at both of its nearest neighbors in the direction perpendicular to the segment DE (Fig. 3.2). Thus, in the case where this setting is incorrect, the 1-point-update values must be replaced by 2-point update values before the point gets “Accepted” in order to propagate the front in order of increase of the true values of T . The question is whether we can guarantee it.

We found examples where indeed a smaller tentative value of T from the 1-point-update was replaced by a larger one from the 2-point-update in a small subset of points. However, numerous numerical experiments showed that such points disappear as we refine the mesh of the input data $s(x_{0i}, T_k)$, $i = 0, 1, \dots, n-1$, $k = 0, 1, \dots, p-1$. Moreover, we did not find any example where the points with 1-point-update values got accepted when they should not be. Thus, although the upwind principle may be violated in theory, we have not found any such example in practice.

3.1.4 Boundary effects

We have input data in the rectangular time domain (x_0, T) , and we look for the output in the rectangular depth domain (x, z) . We will call the image rays arriving at the end points of the “earth surface” segment of the domain the *boundary image rays*. There are three possible behavior of a boundary image ray:

1. the ray is straight, i.e, lies strictly on the boundary of the domain;
2. the ray escapes from the domain;
3. the ray enters the interior of the domain.

If the boundary image ray is either straight or escapes from the domain, then our numerical algorithm computes the values at the boundary mesh points correctly, as the physical domain of dependence of each boundary point lies inside the numerical domain of dependence in these cases. If the boundary ray enters the interior of the domain, then the values at the boundary points are computed by 1-point-updates. The physical domain of dependence for each boundary point lies outside the domain, and hence, cannot be inside the numerical domain of dependence. In this case, our algorithm does not converge in the cone of influence of the boundary points.

3.1.5 Synthetic data examples

Example 1

As a first example, we took the velocity field

$$v(x, z) = 1 + \frac{1}{2} \cos \frac{\pi x}{3} \sin \frac{\pi z}{3},$$

and generated the input data $v(x_0, T)$, $0 \leq x_0 \leq 12$, $0 \leq T \leq 5$ for our time-to-depth conversion algorithm on a $200 \times 200 \, nx_0 \times nT$ mesh by shooting characteristics. We then applied the algorithm to these data and computed the velocity $v(x, z)$ on the $200 \times 400 \, nx \times nz$ mesh. The results are presented in Fig. 3.3-3.4. The exact velocity is shown in Fig. 3.3(a); the input data are shown in Fig. 3.3(b). The velocity found by the algorithm is shown in Fig. 3.3(c). The relative error, i.e., $(v_{found} - v_{exact})/v_{exact}$ is shown in Fig. 3.3(d). The maximal relative error is less than 5 percent and is achieved at the points where the image rays collapse. The image rays computed for the exact velocities are shown in Fig. 3.4. Note that 1) the image rays severely bend, diverge and intersect, and 2) the boundary image rays are straight, which eliminates the errors from the boundary effects.

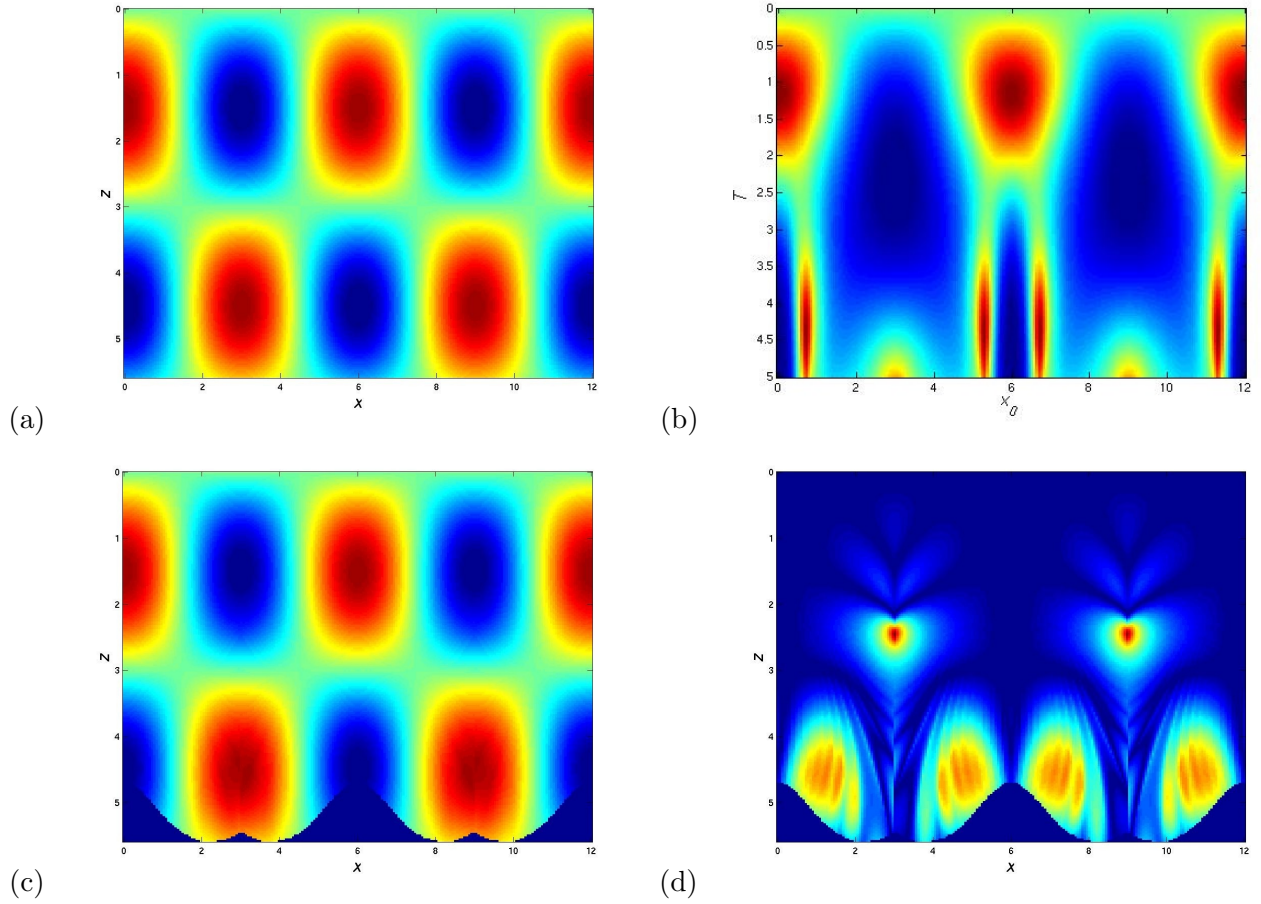


Figure 3.3. (a): The exact velocity $v(x, z) = 1 + \frac{1}{2} \cos \frac{\pi x}{3} \sin \frac{\pi z}{3}$; (b): the input data $v(x_0, T)$; (c): the found velocity $v(x, z)$; (d) the relative error: its maximum is less than 5 percent.

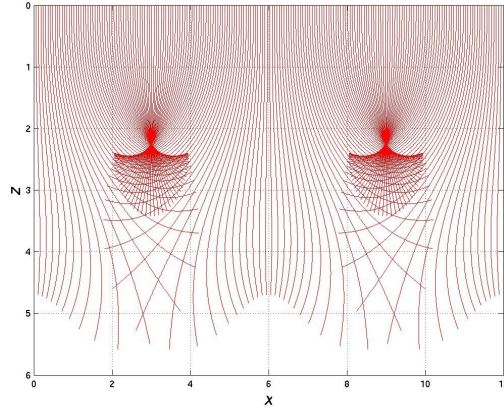


Figure 3.4. The image rays computed for the exact velocity.

Example 2: "Star"

In this section, we consider a velocity field shown in Fig. 3.5 (a). The level sets are shaped as a star with 3 asymmetric tips and there is a Gaussian decay along straight half lines radiating from the geometrical center of the level sets.

More precisely, this velocity field was generated by the construction

$$v(x, z) = 2 + 2 \exp \left(-0.05 \left(\frac{d}{r(\alpha)} \right)^2 \right).$$

Here d is the distance from the geometrical center located at $(x_c = 5, z_c = 5)$. With $\alpha = \arg(x - x_c, z - z_c)$, $r(\alpha)$ is an implicitly defined normalizing function which shapes the level sets of $v(x, z)$:

$$\alpha(\phi) = \phi - 0.25 \cos 3\phi;$$

$$r(\phi) = 1 - 0.5 \cos 3\phi.$$

We obtained the input data $v(x_0, t_0)$ shown in Fig. 3.5 (b) for $0 \leq x_0 \leq 10$ and $0 \leq t_0 \leq 5$ on 1000×1000 mesh by shooting characteristics. Then we applied the time-to-depth algorithm to restore the velocity field $v(x, z)$. The restored velocity is shown in Fig. 3.5 (c). We see that it is very close to the exact velocity. The visible inexactness occurs only within two small areas. The relative error and the image rays are shown in Fig.

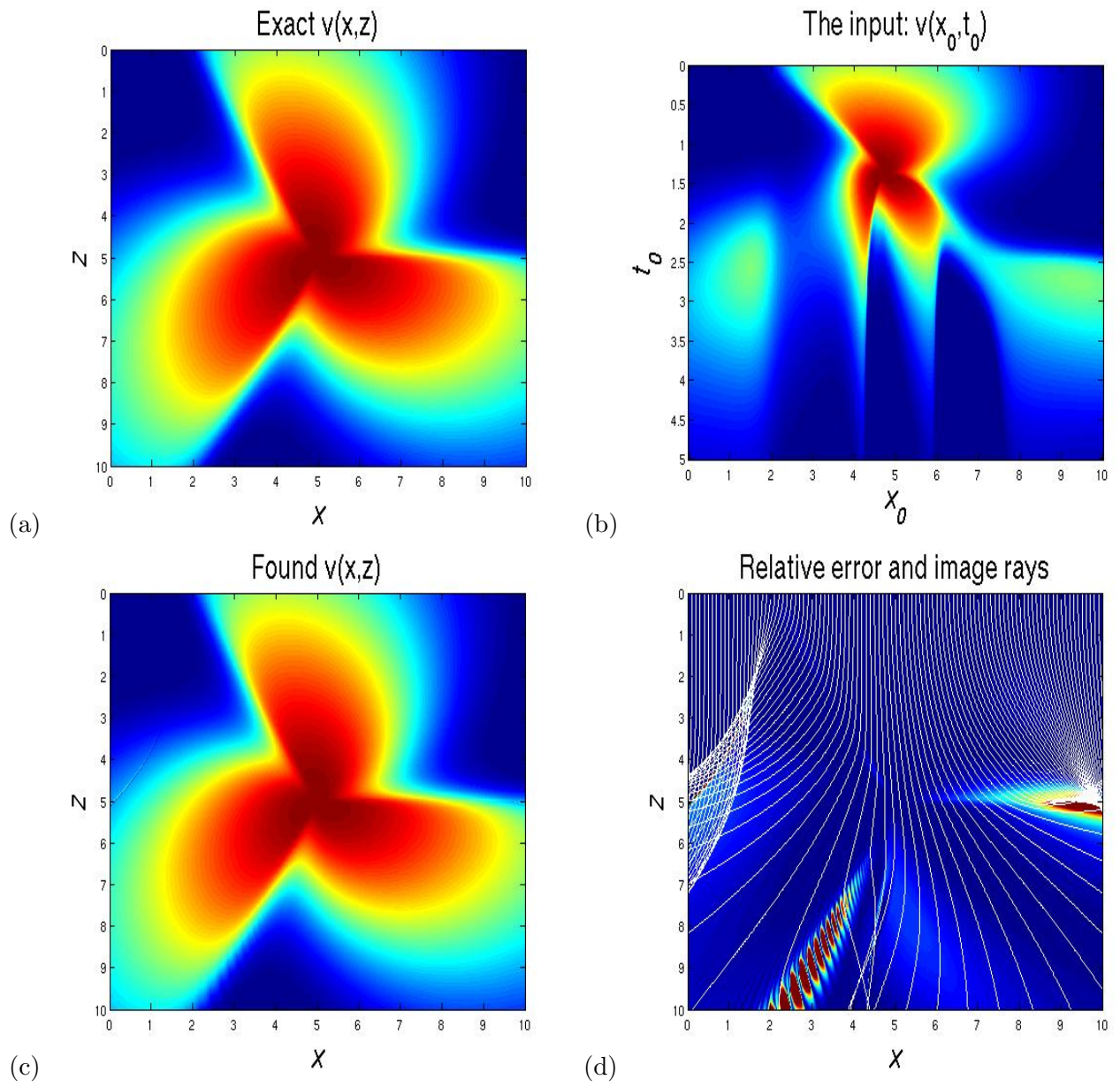


Figure 3.5. (a): The exact velocity $v(x, z)$; (b): the input data $v(x_0, t_0)$; (c): the found velocity $v(x, z)$; (d) the relative error and the image rays.

3.5 (d). Dark red color there corresponds to the areas where the relative error exceeds 1 percent. Such areas are where the image rays either diverge significantly (order of 10) and the gradient of the velocity is large, or along the curve where the image rays coming from different surface points reach at the same time.

Example 3: "Spiral"

Here we consider a modification of the field from the previous example: the Gaussian decay takes place along spiral lines radiating from the geometrical center of the three tip star shaped level sets (Fig. 3.6 (a)).

More precisely, this velocity field is generated as follows. We take a closed curve shaped as a three-tip star given by

$$\begin{aligned}\alpha_0 &= \phi - 0.2 \cos(3\phi); \\ r_0 &= 1 - 0.4 \cos(3\phi).\end{aligned}\tag{3.7}$$

Then, for every point (r_0, α_0) of this curve, consider a spiral line defined by

$$\begin{aligned}r &= r_0 + \sigma; \\ \alpha &= \alpha_0 + 0.11\sigma.\end{aligned}\tag{3.8}$$

The velocity at every point (x, z) is defined by

$$v(x, z) = 2 + 2 \exp \left(-0.1 \left(\frac{d}{r_0} \right)^2 \right).$$

d is the distance from the geometrical center of the level sets located at $(x_c = 5, z_c = 5)$. r_0 is the normalizing factor which is found from equations 7.8, 7.9. We equate $r = d$ and $\alpha = \arg(x - x_c, z - z_c)$ and solve for r_0 , α_0 , ϕ and σ .

The results are shown in Fig. 3.6. The areas where the relative error exceeds one percent (Fig. 3.6(d)) correspond to the dark red color. They occur under the same conditions as they do in the previous example. Once again, the dark red stripe at the right boundary of the domain is due the fact that this area is not covered by the image rays coming from the segment of the surface in hand.

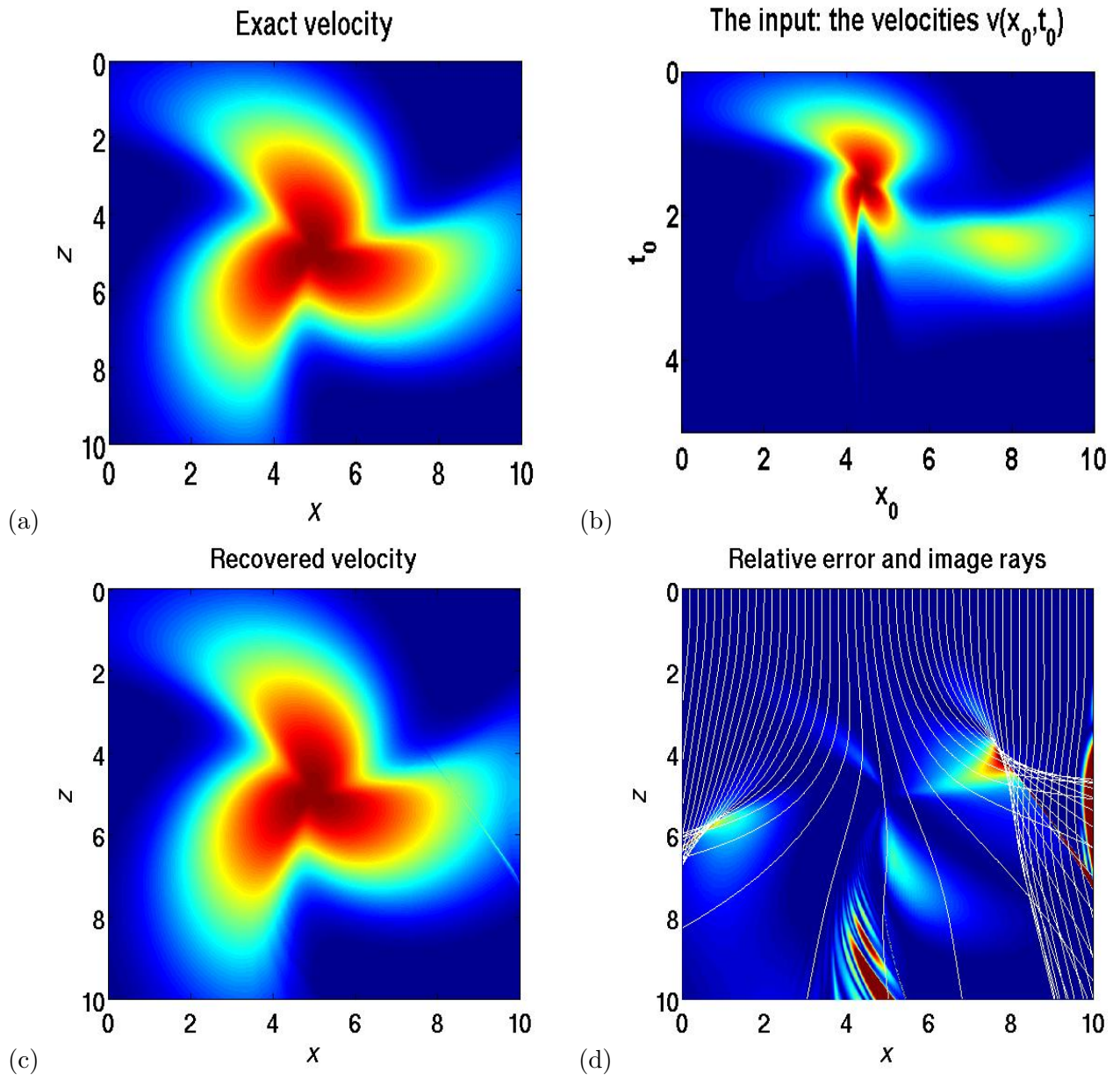


Figure 3.6. (a): The exact velocity $v(x, z)$; (b): the input data $v(x_0, t_0)$; (c): the found velocity $v(x, z)$; (d) the relative error and the image rays.

3.2 Algorithms producing the seismic velocities from the migration velocities

The algorithm introduced in Section 3.1.2 requires the velocities $v(x_0, T)$ as the input: one can use the Dix velocities $v_{Dix}(x_0, T)$ as input. However, as discussed earlier, Dix velocities are obtained with the assumption that the subsurface structures are horizontal and the velocity depends only on the depth. Theorem 2 gave the relation between the Dix velocities and the true seismic velocities. In this section, we introduce two algorithms which try to construct the true seismic velocities from the Dix velocities and use the algorithm in Section 3.1.2 as their essential part.

To be sure, we have just proven that this problem is ill-posed; nonetheless we can develop algorithms which attempt the smoothed reconstruction. The first one is based on the ray tracing approach, and the second one is based on the level set approach. This is a worthwhile endeavor: our numerical examples below demonstrate that the Dix velocities and the true seismic velocities may significantly differ in the case of lateral velocity variation.

3.2.1 Ray tracing approach

The ray tracing algorithm consists of three steps.

Step 1. Find the image rays.

Step 2. Compute the geometrical spreading $|Q| = |\frac{dl}{dx_0}|$ on the image rays and find $v(x_{0i}, T_k)$. Here l is the length of the front.

Step 3. Apply the time-to-depth conversion algorithm from Section 3.1.2 to get $v(x_i, z_j)$, $x_0(x_i, z_j)$ and $t_0(x_i, z_j)$ from $v(x_{0i}, T_k)$.

Let us describe Step 1 in more details. The boundary conditions are $v(x_{0i}, T = 0) = f(x_{0i}, T = 0)$, $Q(x_{0i}, T = 0) = 1$, $P(x_{0i}, T = 0) = 0$. The ray tracing system for the i -th

ray is the following:

$$\begin{aligned}
x_T &= v \sin \theta, & x(0) &= x_{0i}, \\
z_T &= v \cos \theta, & z(0) &= 0, \\
\theta_T &= -v_n = -v_l, & \theta(0) &= 0, \\
Q_T &= v^2 P, & Q(0) &= 1, \\
P_T &= -\frac{v_{nn}}{v} Q = -\left(\frac{v_{ll}}{v} + \frac{\kappa v_T}{v^2}\right) Q, & P(0) &= 0.
\end{aligned} \tag{3.9}$$

Here $v_n = v_x \cos \theta - v_z \sin \theta$ is the derivative of v in the direction normal to the ray (note: $v_n \equiv v_q$); v_l is the derivative of v with respect to the arc length of the front; $v_{nn} = v_{xx} \cos^2 \theta - 2v_{xz} \cos \theta \sin \theta + v_{zz} \sin^2 \theta$ is the second derivative of v in the direction normal to the ray ($v_{nn} \equiv v_{qq}$); v_{ll} is the second derivative of v with respect to the arc length of the front; κ is the curvature of the front. Adalsteinsson and Sethian (Adalsteinsson and Sethian [2002]) derived the following relation between the second derivative of some physical quantity with respect to the arc length of the front and its second derivative along the line tangent to the front:

$$g_{ll} = g_{zz} - (g_x n_x + g_z n_z) \kappa,$$

where n is the unit vector normal to the front. Replacing g with v and noticing that

$$v_x n_x + v_z n_z = v_\tau = \frac{v_T}{v}$$

is the derivative of v with respect to the arc length of the ray, we get the last equation in (3.9).

We solve system (3.9) for all of the rays simultaneously by the forward Euler method as follows.

For $k = 0$ to $k = p - 1$ do:

1. Find the least squares polynomials for the set of points $(l_i, v_i(T_k))$ where l_i is the arc length of the front between ray 0 and ray i at the time T_k , and $v_i(T_k)$ is the value of the velocity on the i -th ray at time T_k . Evaluate $v_l(T_k)$ and $v_{ll}(T_k)$ taking the first and the second derivatives of this polynomial. Moreover, replace the values of the velocity

$v_i(T_k)$ by the values of this polynomial. Evaluate the curvature $\kappa(T_k)$ as follows. Find the least squares polynomials for the sets of points $(i, x_i(T_k))$ and $(i, z_i(T_k))$ where i is the index of the ray, and x_i and z_i are the x - and z -coordinates of the i -th ray at time T_k . Take the first and the second derivatives of these polynomials p_x and p_z and find

$$\kappa = \frac{p'_x p''_z - p'_z p''_x}{(p'^2_x + p'^2_z)^{3/2}}.$$

Approximate $v_T(T_k)$ by

$$v_t(T_k) = \frac{v(T_k) - v(T_{k-1})}{\Delta T}$$

if $k > 0$, and we set $v_T(T_0 = 0) = 0$, since the curvature of the front is zero at $T = 0$.

2. Perform one forward Euler step for each of the rays.
3. For each of the rays find $v_i(T_{k+1}) = f_i(T_{k+1})Q(T_{k+1})$, where $f_i(t_{k+1}) \equiv f(x_{0i}, T_{k+1})$, $i = 0, 1, \dots, n - 1$.

Remarks.

- One can see that we find $v(x_{0i}, T_k)$, $i = 0, 1, \dots, n - 1$, $k = 0, 1, \dots, p - 1$ in the step 1. Hence it is possible immediately go to step 3 to find $v(x_i, z_j)$. However, numerous numerical experiments showed that step 1 computes the image rays $(x(x_{0i}, T_k), z(x_{0i}, T_k))$ significantly more accurately than the velocity $v(x_{0i}, T_k)$. And Step 2 which is very simple, significantly improves the accuracy of $v(x_{0i}, T_k)$.
- As we have shown in Section 2.5.1 the inverse problem is numerically unstable. The use of the least squares polynomials suppresses the growth of the small bumps which naturally appear in result of computations, and hence, stabilizes the algorithm.
- The main limitation of this algorithm is that it blows up as the image rays come too close to each other or diverge too much.
- One can use the additional output $x_0(x_i, z_j)$ and $t_0(x_i, z_j)$ to convert a time-migrated image to depth rather than perform depth migration with the found velocities $v(x_i, z_j)$.

3.2.2 Level set approach

As an alternative to ray tracing, we can formulate a level set approach. The main advantage of this approach, unlike the ray tracing approach, is that it works beyond the first intersection of the image rays, since it tracks the first arrival front.

Level set methods, introduced by Osher and Sethian (Osher and Sethian [1988]), are numerical methods for tracking moving interfaces: they rely in part on the theory of curve and surface evolution given by Sethian (Sethian [1982, 1985]) and on the link between front propagation and hyperbolic conservation laws discussed by Sethian (Sethian [1987]). These techniques recast interface motion as a time-dependent Eulerian initial value partial differential equation. For a general introduction and overview, see Sethian (Sethian [1999B]).

The main idea of a level set method is the representation of a front as the zero level set of some higher dimensional function. In our context, we want to propagate the wave front coinciding with the flat surface at $t = 0$ downward the earth. We embed the wave front into a 2D function $\phi(x, z)$ so that the front is its zero level set. Furthermore, we embed the quantities Q and P defined on the front into 2D functions $q(x, z)$ and $p(x, z)$ so that at each moment of time $Q = q(x, z)|_{\{(x, z)|\phi(x, z)=0\}}$ and $P = p(x, z)|_{\{(x, z)|\phi(x, z)=0\}}$, i.e., Q and P coincide with q and p on the zero level set of $\phi(x, z)$. Let

$$g_x = \frac{\phi_x}{|\nabla\phi|}, \quad g_z = \frac{\phi_z}{|\nabla\phi|}.$$

Let us find the system of equations for q and p . First note that

$$v_{nn} = v_{xx} \cos^2 \theta - 2v_{xz} \cos \theta \sin \theta + v_{zz} \sin^2 \theta.$$

Second, at each point of the the zero level set of ϕ , i.e. at each front point,

$$g_x = \cos \theta, \quad g_z = \sin \theta.$$

Then we get the following equations for q and p :

$$q_t = v^2 p, \quad p_t = -\frac{v_{xx}g_x^2 - 2v_{xz}g_x g_z + v_{zz}g_z^2}{v} q. \quad (3.10)$$

These equations coincide with the equations for Q and P on the front. Here, we switch the notation for time from T to t . We will reserve the notation T for auxiliary times in the fast marching parts of our level set algorithm.

Thus, we have to solve the following system of PDE's:

$$\begin{aligned}\phi_t + v(x, z)|\nabla\phi| &= 0, \\ q_t &= v^2(x, z)p, \\ p_t &= -\frac{v_{xx}g_x^2 - 2v_{xz}g_xg_z + v_{zz}g_z^2}{v(x, z)}q.\end{aligned}\tag{3.11}$$

As before, we have the input data $f(x_0, t) = \frac{v(x_0, t)}{|Q(x_0, t)|}$ given in (x_0, t) space on a $n \times k$ mesh, and we need to obtain $v(x, z)$ in (x, z) space on a $n \times m$ mesh.

Initialization: Set $q(x, z) = 1$, $p(x, z) = 0$, which is correct for the front at $t = 0$. Set $v(x, 0) = f(x_0, 0)$ and attach labels "x" to the surface points. Set $\phi(x, z) = z$, i.e., make the level set function a signed distance function.

We solve system (3.11) in the following time cycle: for $k = 0$ to $p - 1$ do:

1. Starting with the current "x" points, solve the system

$$q(x, z)|\nabla T| = \frac{1}{f(x_0, T)}, \quad \nabla x_0 \cdot \nabla T = 0$$

using the Fast Marching time-to-depth conversion algorithm introduced in Section 3.1.2 to find $v(x, z) = f(x, z)q(x, z)$ for the current $q(x, z)$.

2. Attach labels "x" to the accepted points for which T is not greater than the current value of time t_k .
3. Detect the zero level set of ϕ . Find the velocity v at the zero level set of ϕ and build an extension of v solving the system

$$|\nabla d| = 1, \quad \nabla d \cdot \nabla v_{ext} = 0,$$

with the boundary conditions $d = 0$ and $v_{ext} = v$ at the zero level set of ϕ , using the Fast Marching Method, as it is suggested in Sethian [1996, 1999A]. If the extended velocity is built this way, an initial signed distance function ϕ remains so as it evolves.

4. Perform a time step: Compute the quantities g_x and g_z for the current ϕ . Find v_{xx} , v_{xz} and v_{zz} by finding least square polynomials for each grid line $x = x_i$ and $z = z_j$

and evaluating their derivatives. Make one forward Euler step for equations (3.10) to find new q and p . Solve the level set equation

$$\phi_t + v_{ext}|\nabla\phi| = 0$$

from $t = k\Delta t$ to $t = (k+1)\Delta t$ by the forward Euler method with a time step satisfying the CFL condition.

We stress that the main advantage of this algorithm in comparison with the ray tracing algorithm is that it can work even if the image rays intersect, since it tracks the first arrival front.

Having obtained the true seismic velocities $v(x_i, z_j)$ one can perform depth migration to obtain an improved seismic image in the Cartesian coordinates. Alternatively, knowing the velocity $v(x_i, z_j)$ one can apply Sethian's fast marching method Sethian [1996] to obtain $t_0(x_i, z_i)$ and $x_0(x_i, z_i)$ to convert the time migrated image to depth.

Chapter 4

Synthetic data examples in 2D

4.1 Example 1

The example in this section allows us to compare performances of the ray tracing algorithm and level set algorithm with a somewhat typical approach. One typical approach to seismic velocity estimation is to compute the Dix velocities and then apply image ray tracing. Here we will replace the image ray tracing with our time-to-depth conversion algorithm.

We considered the velocity fields of the form:

$$v(x, z) = 1 + \exp(-c(x^2 + (z - 1)^2)), \quad x_0 \in [-2, 2], \quad t \in [0, 0.7]. \quad (4.1)$$

We took $c = 0.5$, $c = 1$ and $c = 1.5$. The larger c , the sharper the Gaussian anomaly. For each of these fields we created the input data $f(x_0, t)$ on a 200×200 $x_0 \times t$ mesh and applied each of the three algorithms to them: the time-to-depth conversion, the ray tracing, and the level set. The output $v(x, z)$ is given on 200×200 $x \times z$ mesh.

The exact velocity, the input data (the Dix velocity), the found velocity and the image rays for the sharpest Gaussian anomaly corresponding $c = 1.5$ are shown in Fig. 4.1. We see that the Dix velocity qualitatively differs from the exact velocity and the found velocity resembles the exact velocity much more closely than the Dix velocity.

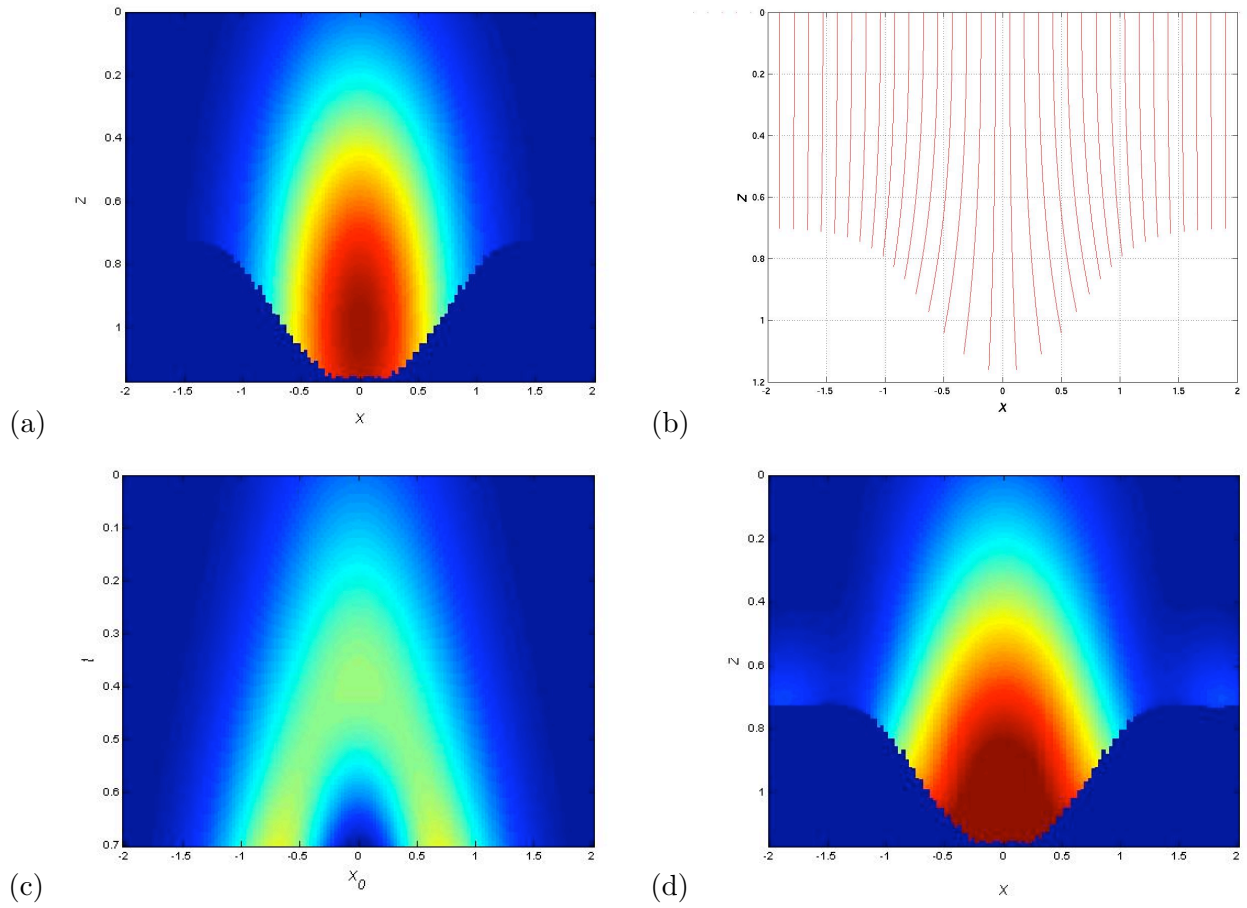


Figure 4.1. (a): the exact velocity $v(x, z)$; (b) the image rays; (c): the input data $f(x_0, t) \equiv v_{Dix}(x_0, t)$; (d): the found velocity $v(x, z)$.

The results are summarized in Table (4.1). We see that

Table 4.1. The maximal relative errors produced by the time-to-depth conversion, the ray tracing and the level set algorithms on the data from the velocity field (4.1).

Algorithm	<i>Time-to-depth</i>	<i>Ray tracing</i>	<i>Level set</i>
$c = 0.5$	0.31	0.023	0.078
$c = 1$	0.44	0.11	0.079
$c = 1.5$	0.49	0.29	0.20

- the ray tracing and the level set produce significantly more accurate results than the typical approach;
- the ray tracing approach is more accurate than the level set where the image rays diverge moderately, while it becomes less accurate as the divergence of the image rays increases.

Note that if the image rays diverge severely so that the derivative v_{nn} (or, in different notations, v_{qq}) becomes large, both our ray tracing and level set algorithms blow up, while the time-to-depth convergence algorithm produces inaccurate but stable results.

4.2 Example 2

In this section we also consider an example with a Gaussian anomaly, but with numbers closer to real seismic data:

$$v(x, z) = 2 + 2 \exp \left(-(x^2 + (z - 2)^2) \right),$$

$$x_0 \in [-3, 3], \quad t \in [0, 1],$$

The center of the anomaly lies at the depth of 2 km and the background velocity is 2 km/sec. The results (Fig. 4.2) are produced by the level set algorithm. The found velocity resembles the exact velocity while the Dix velocity and the found velocity differ qualitatively.

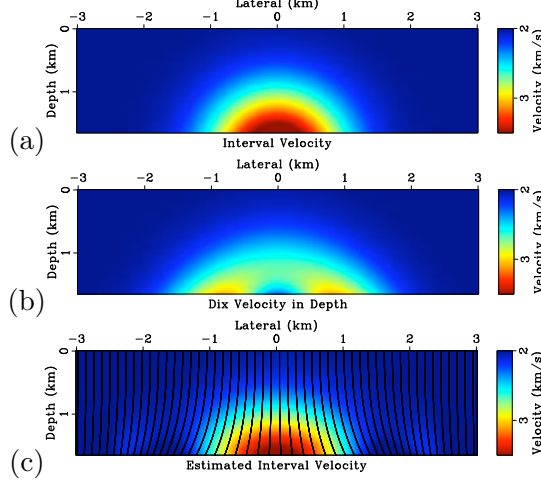


Figure 4.2. (a): the exact velocity $v(x, z)$; (b) the Dix velocity converted to depth by "vertical stretch"; (c): the found velocity $v(x, z)$ and the image rays.

4.3 Example 3

In this section, we consider the upper part of the velocity field of the type as in Example 2: "Star" in Section 3.1.5. This velocity field was generated as follows:

$$v(x, z) = 2 + 2 \exp \left(-0.1 \left(\frac{d}{r(\alpha)} \right)^2 \right).$$

Here d is the distance from the geometrical center located at $(x_c = 5, z_c = 5)$. $\alpha = \arg(x - x_c, z - z_c)$, and $r(\alpha)$ is an implicitly defined normalizing function which shapes the level sets of $v(x, z)$:

$$\alpha(\phi) = \phi - 0.1 \cos 3\phi;$$

$$r(\phi) = 1 - 0.2 \cos 3\phi.$$

The input data were generated on 150×150 (x_0, t_0) mesh for $0 \leq x_0 \leq 10$ km, $0 \leq t_0 \leq 1.5$ sec. The output is shown on a 150×75 (x, z) mesh up to 2.33 km in depth.

The exact velocity, the Dix velocity converted to depth by our time-to-depth algorithm, the recovered velocity and the image rays are shown in Fig. 4.3 (a), (b), (c) respectively.

Comparing the differences between Figures 4.3 (a) and (b) and between Figures 4.3 (a)

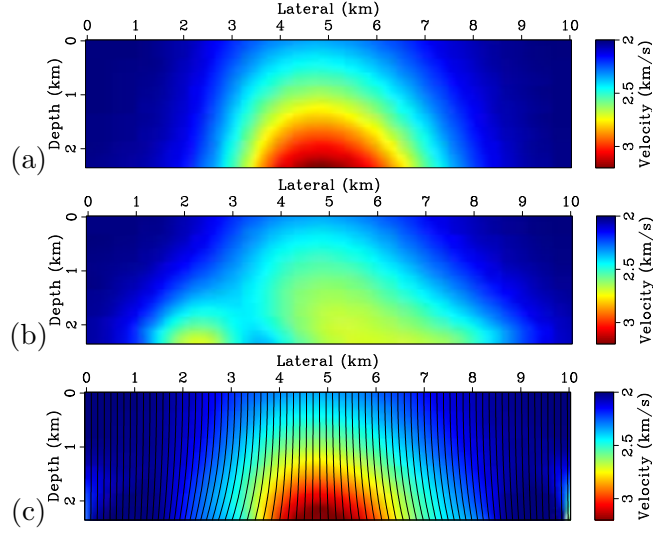


Figure 4.3. (a): The exact velocity $v(x, z)$; (b): the Dix velocity converted to depth; (c): the found velocity $v(x, z)$ and the image rays.

and (c) we see that the velocity recovered using our level set algorithm is close to the exact velocity while the Dix velocity converted to depth differs qualitatively from it.

Chapter 5

Field data examples

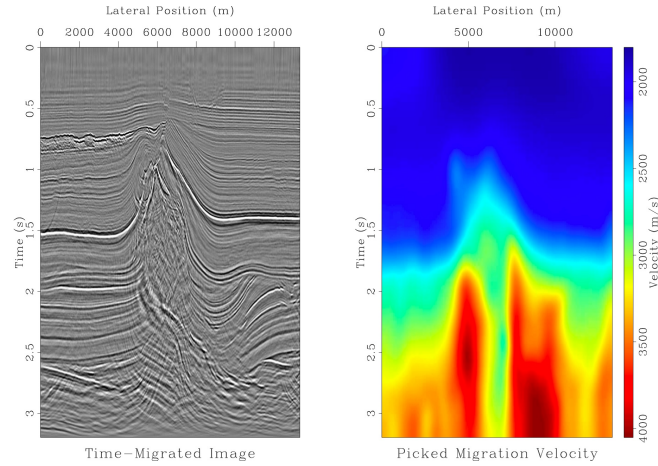


Figure 5.1. Left: seismic image from North Sea obtained by prestack time migration using velocity continuation Fomel [2003]. Right: the corresponding time migration velocity.

In this section we consider a field data example coming from the North Sea (Fig. 5.1, left). The main feature in this image is the salt dome. Typically, the velocity inside the salt is higher than it is in the surrounding rock. Salt is light and it pushes the layers up as it comes from inside the earth. The lateral velocity variation here is severe according to typical geophysical situations. Note rapidly changing values inside the salt dome, which indicate that the lateral velocity variation is too large for the time migration.

In Fig. 5.1, right, the time migration velocities chosen in the process of making this im-

age are shown. Using these time migration velocities, the Dix velocities were then obtained and smoothed. The level set algorithm was then applied to these Dix velocities to estimate seismic velocities $v(x, z)$. These seismic velocities together with the image rays computed from the by shooting characteristics are shown in Fig. 5.2). The depth domain (x, z) was cut at 3.3 km to make the found $v(x, z)$ into a rectangular matrix.

One can compare the smoothed Dix velocities and the found seismic velocities (Fig. 5.3) and see that they differ significantly starting from about 1 km in depth.

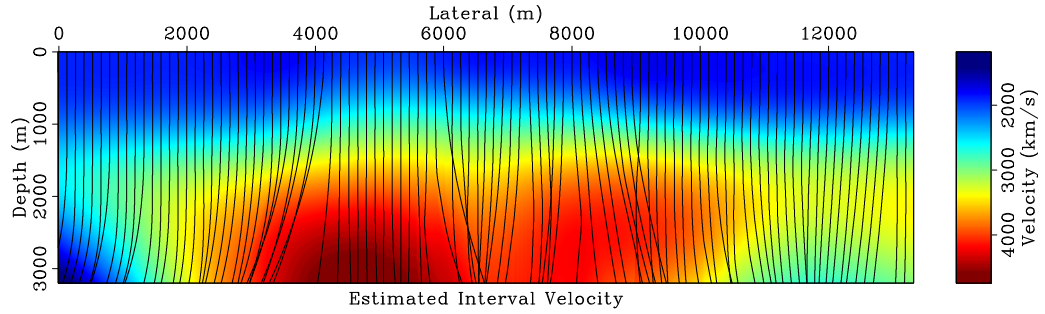


Figure 5.2. The found seismic velocity $v(x, z)$ and the image rays computed from it.

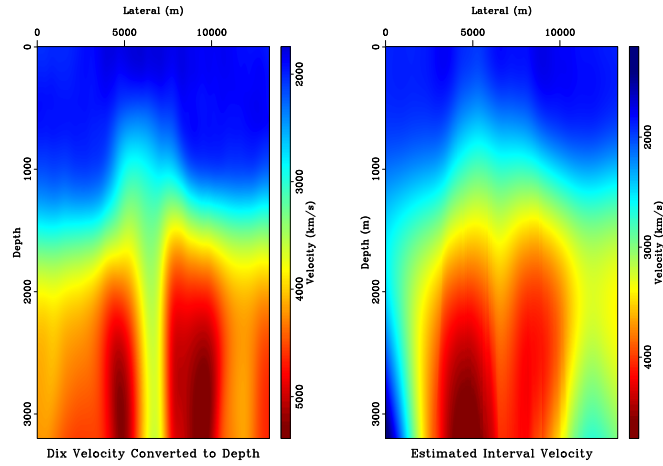


Figure 5.3. The smoothed Dix velocity $v_{Dix}(x_0, t_0)$ (left) vs the found seismic velocity $v(x, z)$ (right).

The depth migrated image, built using the calculated $v(x, z)$, is shown in Fig. 5.4 (a).

The image is in the regular Cartesian coordinates. It shows subsurface structures up to 3.3 km in depth which is quite deep according to geophysical standards. There is a noisy reconstruction inside the salt dome but the surrounding layers are resolved well. Overall, this image looks reasonable.

We applied Sethian's Fast Marching Method to solve the Eikonal Equation with the found velocity $v(x, z)$ and found the matrices $t_0(x, z)$ and $x_0(x, z)$. Then we converted the time migrated image in Fig. 5.1, (left) to depth values using these matrices. The resulting image is shown in Fig. 5.4 (b). Comparing the two images in depth in Fig. 5.4 obtained in these two alternative ways, we see a good agreement between them.

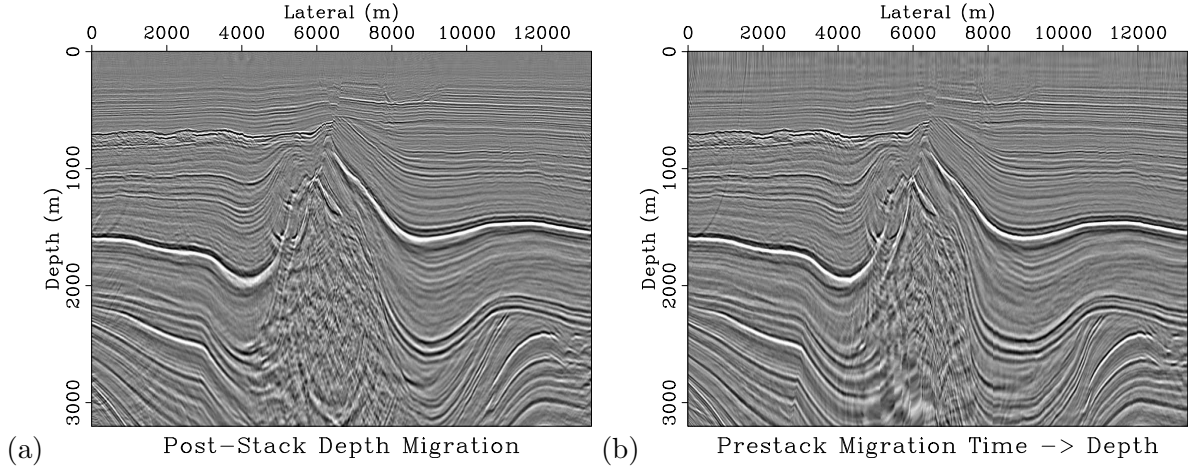


Figure 5.4. (a) The poststack depth migrated image obtained with the found $v(x, z)$; (b) The prestack time migrated image converted to depth.

For comparison, we also used the Dix velocities to perform the depth migration. The resulting image is shown in Fig. 5.5, left, while the results of the depth migration with the velocities found by our level set algorithm are shown in Fig. 5.5, right. There is a visible change in the lower part of the image and several indications that the change is in the right direction.

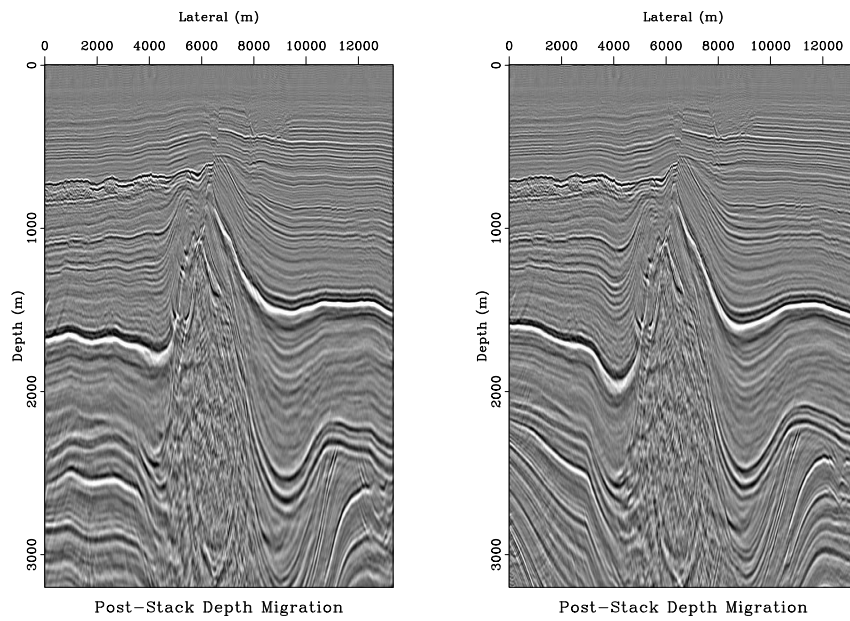


Figure 5.5. The poststack depth migration using the Dix velocities (left) vs the poststack depth migration using the estimated seismic velocities (right).

Chapter 6

Numerical algorithms in 3D

In this section, we present 3D versions of our algorithms. We present a 3D Dijkstra-like fast method for time-to-depth conversion, and a 3D Ray tracing approach for the inverse problem. A 3D level set version is underway.

In more detail, we present a ray tracing approach for solving the inverse problem in 3D (see Section 2.5.2). This approach is the extension of the ray tracing approach for 2D (see Section 3.2.1). The input data are the set of matrices

$$\mathbf{F}(x_0, y_0, t_0) \equiv \frac{\partial \mathbf{K}(x_0, y_0, t_0)}{\partial t_0} = v^2(x, y, z)(\mathbf{Q}^T(x_0, y_0, t_0)\mathbf{Q}(x_0, y_0, t_0))^{-1}. \quad (6.1)$$

Here $v(x, y, z)$ is the seismic velocity at the location reached by the image being traced backwards for time t_0 starting from the surface point (x_0, y_0) , and $\mathbf{Q}(x_0, y_0, t_0)$ corresponds to the telescopic family of rays traced along with the image ray for time t_0 . The input data are given on the 3D time domain mesh (x_{0i}, y_{0j}, t_{0k}) , $i = 0, \dots, n_x - 1$, $j = 0, \dots, n_y - 1$, $k = 0, \dots, n_t - 1$, $x_{min} = x_{00} \leq x_{0i} \leq x_{0, n_x - 1} = x_{max}$, $y_{min} = y_{00} \leq y_{0j} \leq y_{0, n_y - 1} = y_{max}$, $0 = t_0 \leq t_k \leq t_{n_t - 1} = t_{max}$. The output is the four sets of data $v(x, y, z)$, $t_0(x, y, z)$, $x_0(x, y, z)$ and $y_0(x, y, z)$ given on the 3D depth domain mesh (x_i, y_j, z_k) , $i = 0, \dots, n_x - 1$, $j = 0, \dots, n_y - 1$, $k = 0, \dots, n_z - 1$, $x_{min} = x_0 \leq x_i \leq x_{n_x - 1} = x_{max}$, $y_{min} = y_0 \leq y_j \leq y_{n_y - 1} = y_{max}$, $0 = z_0 \leq z_k \leq z_{n_z - 1} = z_{max}$.

This approach consists of the following three steps.

Step 1. Ray tracing algorithm which computes the image rays the image rays.

Step 2. Using the image rays found in Step 1 compute the geometrical spreading which equals $|\det Q|$ (Popov [2002]) and determine the velocity $v(x_0, y_0, t_0)$ from the input data 6.1.

Step 3. Convert the velocities $v(x_0, y_0, t_0)$ given in the time coordinates (x_0, y_0, t_0) to depth: find $v(x, y, z)$.

Now let us describe each of these three steps in details.

6.1 Step 1: Ray tracing algorithm

6.1.1 The ray equation in the spherical coordinates

At each moment of time t , we will describe a ray by the following five coordinates: $x(t)$, $y(t)$, $z(t)$, $\theta(t)$, $\phi(t)$, where θ is the angle between the direction of the ray and the x -axis, and ϕ is the angle between the projection of the ray onto yz -plane and the positive direction of the y -axis. We start to trace the image rays from the surface normal to it. Our choice of angles allows us to avoid (or, at least, substantially postpone) the problem of ϕ being undefined where $\theta = 0$. As before, define the Hamiltonian by

$$H = \frac{1}{2} (|\nabla t|^2 - s^2(x, y, z)) \equiv \frac{1}{2} (|\vec{p}|^2 - s^2(x, y, z)) \equiv 0, \quad (6.2)$$

where s is the slowness - the reciprocal of the velocity. Then the ray equations in the Hamiltonian form are

$$\begin{aligned} \frac{dx}{d\sigma} &= p_x, & \frac{dp_x}{d\sigma} &= ss_x, \\ \frac{dy}{d\sigma} &= p_y, & \frac{dp_y}{d\sigma} &= ss_y, \\ \frac{dz}{d\sigma} &= p_z, & \frac{dp_z}{d\sigma} &= ss_z. \end{aligned} \quad (6.3)$$

Then we introduce

$$\theta = \arctan \frac{\sqrt{p_y^2 + p_z^2}}{p_x}, \quad \phi = \arctan \frac{p_z}{p_y}, \quad (6.4)$$

and perform the variable change in system 6.3.

$$\begin{aligned}\frac{d\theta}{d\sigma} &= \frac{d\theta}{dp_x} \frac{dp_x}{d\sigma} + \frac{d\theta}{dp_y} \frac{dp_y}{d\sigma} + \frac{d\theta}{dp_z} \frac{dp_z}{d\sigma} \\ &= \frac{1}{1 + \frac{p_y^2 + p_z^2}{p_x^2}} \left(-\frac{\sqrt{p_y^2 + p_z^2}}{p_x^2} s s_x + \frac{p_y}{p_x \sqrt{p_y^2 + p_z^2}} s s_y + \frac{p_z}{p_x \sqrt{p_y^2 + p_z^2}} s s_z \right).\end{aligned}\quad (6.5)$$

Then, using the relations

$$p_x = s \cos \theta, \quad p_y = s \sin \theta \cos \phi, \quad p_z = s \sin \theta \sin \phi, \quad p_y^2 + p_z^2 = s^2 \sin^2 \theta, \quad (6.6)$$

we proceed with equation 6.5:

$$\begin{aligned}\frac{d\theta}{d\sigma} &= \cos^2 \theta \left(-\frac{\sin \theta}{\cos^2 \theta} s_x + \frac{\sin \theta \cos \phi}{\cos \theta \sin \theta} s_y + \frac{\sin \theta \sin \phi}{\cos \theta \sin \theta} s_z \right) \\ &= -s_x \sin \theta + s_y \cos \theta \cos \phi + s_z \cos \theta \sin \phi.\end{aligned}\quad (6.7)$$

Compute the derivative of ϕ .

$$\begin{aligned}\frac{d\phi}{d\sigma} &= \frac{d\phi}{dp_y} \frac{dp_y}{d\sigma} + \frac{d\phi}{dp_z} \frac{dp_z}{d\sigma} \\ &= \frac{1}{1 + \frac{p_z^2}{p_y^2}} \left(-\frac{p_z}{p_y^2} s s_y + \frac{1}{p_y} s s_z \right) \\ &= \cos^2 \phi \left(-\frac{\sin \theta \sin \phi}{\sin^2 \theta \cos^2 \phi} s_y + \frac{1}{\sin \theta \cos \phi} s_z \right) \\ &= \frac{-s_y \sin \phi + s_z \cos \phi}{\sin \theta}.\end{aligned}\quad (6.8)$$

Now recall that $\frac{dt}{d\sigma} = s^2 = \frac{1}{v^2}$ and $\frac{ds}{d\alpha} = -\frac{1}{v^2} \frac{dv}{d\alpha}$ and switch to t as an independent variable.

Then the ray tracing system is:

$$\begin{aligned}\frac{dx}{dt} &= v \cos \theta, \\ \frac{dy}{dt} &= v \sin \theta \cos \phi, \\ \frac{dz}{dt} &= v \sin \theta \sin \phi, \\ \frac{d\theta}{dt} &= v_x \sin \theta - v_y \cos \theta \cos \phi - v_z \cos \theta \sin \phi \equiv \nabla v \cdot \vec{l}_1, \\ \frac{d\phi}{dt} &= \frac{1}{\sin \theta} (v_y \sin \phi - v_z \cos \phi) \equiv \frac{1}{\sin \theta} (\nabla v \cdot \vec{l}_2),\end{aligned}\quad (6.9)$$

where

$$\vec{l}_1 \equiv \begin{pmatrix} \sin \theta \\ -\cos \theta \cos \phi \\ -\cos \theta \sin \phi \end{pmatrix}, \quad \vec{l}_2 \equiv \begin{pmatrix} 0 \\ \sin \phi \\ -\cos \phi \end{pmatrix}.\quad (6.10)$$

Along with the rays themselves we need to trace the 2×2 matrices \mathbf{Q} and \mathbf{P} for the telescopic families of rays around each of the image rays. As it was shown in Popov [2002], Červený [2001], these matrices evolve according to

$$\frac{d\mathbf{Q}}{dt} = v^2 \mathbf{P}, \quad \frac{d\mathbf{P}}{dt} = -\frac{1}{v} \mathbf{V} \mathbf{P}. \quad (6.11)$$

The matrix $\mathbf{V} = (v_{q_i q_j})_{i,j=1,2}$ is the matrix of the second derivatives of $v(x, y, z)$ along the directions \vec{e}_1 and \vec{e}_2 , normal to the central rays (the image rays in our case), which evolve as follows:

$$\frac{d\vec{e}_1}{dt} = v_{q_1} \bigg|_{\substack{q_1=0 \\ q_2=0}} \vec{\tau}, \quad \frac{d\vec{e}_2}{dt} = v_{q_2} \bigg|_{\substack{q_1=0 \\ q_2=0}} \vec{\tau}, \quad (6.12)$$

where $\vec{\tau}$ is a unit vector tangent to the image ray.

Thus, for each image ray we need to trace the following system of 19 equations

$$\begin{aligned} \frac{dx}{dt} &= v \cos \theta, \\ \frac{dy}{dt} &= v \sin \theta \cos \phi, \\ \frac{dz}{dt} &= v \sin \theta \sin \phi, \\ \frac{d\theta}{dt} &= \nabla v \cdot \vec{l}_1, \\ \frac{d\phi}{dt} &= \frac{1}{\sin \theta} (\nabla v \cdot \vec{l}_2), \\ \frac{dQ_{11}}{dt} &= v^2 P_{11}, \quad \frac{dQ_{12}}{dt} = v^2 P_{12}, \\ \frac{dQ_{21}}{dt} &= v^2 P_{11}, \quad \frac{dQ_{22}}{dt} = v^2 P_{22}, \\ \frac{dP_{11}}{dt} &= -\frac{1}{v} (v_{q_1 q_1} Q_{11} + v_{q_1 q_2} Q_{21}), \quad \frac{dP_{12}}{dt} = -\frac{1}{v} (v_{q_1 q_1} Q_{12} + v_{q_1 q_2} Q_{22}), \\ \frac{dP_{21}}{dt} &= -\frac{1}{v} (v_{q_2 q_1} Q_{11} + v_{q_2 q_2} Q_{21}), \quad \frac{dP_{22}}{dt} = -\frac{1}{v} (v_{q_2 q_1} Q_{12} + v_{q_2 q_2} Q_{22}), \\ \frac{de_{1x}}{dt} &= v_{q_1} \bigg|_{\substack{q_1=0 \\ q_2=0}} \cos \theta, \quad \frac{de_{2x}}{dt} = v_{q_2} \bigg|_{\substack{q_1=0 \\ q_2=0}} \cos \theta, \\ \frac{de_{2y}}{dt} &= v_{q_1} \bigg|_{\substack{q_1=0 \\ q_2=0}} \sin \theta \cos \phi, \quad \frac{de_{2y}}{dt} = v_{q_2} \bigg|_{\substack{q_1=0 \\ q_2=0}} \sin \theta \cos \phi, \\ \frac{de_{1z}}{dt} &= v_{q_1} \bigg|_{\substack{q_1=0 \\ q_2=0}} \sin \theta \sin \phi, \quad \frac{de_{2z}}{dt} = v_{q_2} \bigg|_{\substack{q_1=0 \\ q_2=0}} \sin \theta \sin \phi \end{aligned} \quad (6.13)$$

where \vec{l}_1 and \vec{l}_2 are defined by equation (6.10). The initial conditions are

$$\begin{aligned} x(0) &= x_{0i}, \quad y(0) = y_{0j}, \quad z(0) = 0, \quad \theta(0) = \frac{\pi}{2}, \quad \phi(0) = \frac{\pi}{2}, \\ \mathbf{Q}(0) &= I_2, \quad \mathbf{P}(0) = \mathbf{0}, \\ \vec{e}_1(0) &= (1, 0, 0)^T, \quad \vec{e}_2(0) = (0, 1, 0)^T. \end{aligned} \tag{6.14}$$

6.1.2 The ray tracing algorithm

We need to solve system (6.13) under the circumstance that the velocity $v(x, y, z)$ is unknown; instead, we are given the input data $F(x_0, y_0, t_0)$, equation (6.1) given on the time domain mesh. First note that since the vectors \vec{e}_1 and \vec{e}_2 are orthogonal to the image rays, the directions along which the derivatives $v_{q_i q_j}$ and v_{q_i} , $i, j = 1, 2$ are evaluated, are tangent to the front. We can easily show that the vectors \vec{l}_1 , and \vec{l}_2 are also orthogonal to the image rays and hence tangent to the front. Indeed,

$$\begin{aligned} \vec{l}_1 \cdot \vec{\tau} &= \begin{pmatrix} \sin \theta \\ -\cos \theta \cos \phi \\ -\cos \theta \sin \phi \end{pmatrix} \cdot \begin{pmatrix} \cos \theta \\ \sin \theta \cos \phi \\ \sin \theta \sin \phi \end{pmatrix} = 0, \\ \vec{l}_2 \cdot \vec{\tau} &= \begin{pmatrix} 0 \\ \sin \phi \\ -\cos \phi \end{pmatrix} \cdot \begin{pmatrix} \cos \theta \\ \sin \theta \cos \phi \\ \sin \theta \sin \phi \end{pmatrix} = 0. \end{aligned}$$

Thus, all of the directions along which we need to evaluate the derivatives of the velocity are tangent to the front.

It was shown by Adalsteinsson and Sethian [2002] that for a flat curve the following relation takes place:

$$v_{ss} = v_{qq} - v_{\tau} \kappa, \tag{6.15}$$

where v_{ss} is the second derivative along the curve, v_{qq} is the second derivative along the tangent line, v_{τ} is the derivative along the normal direction to the curve, and κ is the curvature of the curve. The relation (6.15) is valid for a nonflat curve as well, and the proof is identical:

Proof Let $(x(s), y(s), z(s))$ be a curve, and l be the natural parameter along it - the arc length. Let $\vec{e} = \left(\frac{dx}{ds}, \frac{dy}{ds}, \frac{dz}{ds}\right)$ be its velocity vector, which is the unit vector tangent to it. Then the curvature κ is $\left|\left(\frac{d^2x}{ds^2}, \frac{d^2y}{ds^2}, \frac{d^2z}{ds^2}\right)\right|$. Differentiate v twice with respect to the arc length:

$$\begin{aligned} v_{ss} &= \frac{\partial}{\partial s} \left(\frac{\partial}{\partial s} v(x(s), y(s), z(s)) \right) = \frac{\partial}{\partial s} (v_x e_x + v_y e_y + v_z e_z) \\ &= \frac{\partial}{\partial s} (\nabla v \cdot \vec{e}) = \left(\frac{\partial}{\partial s} \nabla v \right) \cdot \vec{e} + \nabla v \cdot \frac{\partial \vec{e}}{\partial s} \\ &= (\vec{e})^T D^2 v \vec{e} - \kappa \nabla v \cdot \vec{n} = v_{qq} - v_\tau \kappa, \end{aligned}$$

as we wanted to prove. Here $D^2 v$ is the matrix of the second derivatives of v .

Now we are ready to present the ray tracing algorithm. Since the matrix \mathbf{Q} at the surface is the identity matrix, $\mathbf{F}(x_0, y_0, 0) = \begin{pmatrix} v^2 & 0 \\ 0 & v^2 \end{pmatrix}$, we can evaluate the velocity at the surface by

$$v(x_0, y_0, 0) = \sqrt[4]{\mathbf{F}_{11} \mathbf{F}_{22}}. \quad (6.16)$$

Then we make the first forward Euler time step for system (6.13) using the initial conditions (6.14) and taking into account that $x \equiv q_1$ and $y \equiv q_2$ at the surface. We then find the velocity at the next moment of time t_1 by

$$v(x_0, y_0, t_1) = \sqrt[4]{\det \mathbf{F}(x_0, y_0, t_1) (\det \mathbf{Q}(x_0, y_0, t_1))^2}.$$

The further time steps are given by the following:

For $k = 1$ to $n_t - 2$ do

1. Estimate the curvatures of the grid curves (x_0, y_{0j}, t_k) , $j = \text{const}$, and (x_{0i}, y_0, t_k) , $i = \text{const}$ and estimate the first and the second derivatives of the velocity along the tangent lines to these curves. We first approximate the functions $x(x_0, y_{0j}, t_k)$, $y(x_0, y_{0j}, t_k)$ and $z(x_0, y_{0j}, t_k)$ by a least squares polynomial and find its second derivatives with respect to the arc length s_1 . Then approximate the velocity along these lines using a least squares polynomial and evaluate its the first and the second derivatives with respect to the arc length s_1 . Correct the second derivatives using formula (6.15).

Repeat this procedure for the grid curves (x_{0i}, y_0, t_k) and also find the mixed second derivative of v $\frac{\partial^2 v}{\partial s_1 \partial s_2}$.

2. We have estimated the first and the second derivatives of v along the tangent lines to the grid curves: v_{s_1} , v_{s_2} , $v_{s_1 s_1}$, $v_{s_2 s_2}$ and $v_{s_1 s_2}$. We need to find the derivatives of the velocity along the directions \vec{e}_1 , \vec{e}_2 , \vec{l}_1 and \vec{l}_2 . As we have shown, all of these directions are tangent to the front, hence they lie in the same plane as the directions s_1 and s_2 . We express these directions in terms of s_1 and s_2 using the least squares and then find the needed derivatives. Let \vec{s}_1 and \vec{s}_2 be unit vectors in the directions s_1 and s_2 . Let

$$\vec{e}_1 = b_{11}\vec{s}_1 + b_{12}\vec{s}_2, \quad (6.17)$$

$$\vec{e}_2 = b_{21}\vec{s}_1 + b_{22}\vec{s}_2,$$

therefore,

$$v_{q_1} = \vec{e}_1 \cdot \nabla v = (b_{11}\vec{s}_1 + b_{12}\vec{s}_2) \cdot \nabla v = b_{11}v_{s_1} + b_{12}v_{s_2}, \quad (6.18)$$

$$v_{q_2} = \vec{e}_2 \cdot \nabla v = (b_{21}\vec{s}_1 + b_{22}\vec{s}_2) \cdot \nabla v = b_{21}v_{s_1} + b_{22}v_{s_2}.$$

Similarly we find $\vec{l}_i \cdot \nabla v$, $i = 1, 2$. We compute the matrix $(v_{q_i q_j})_{i,j=1,2}$ as follows.

$$\begin{aligned} v_{q_i q_j} &= (\vec{e}_i)^T D^2 v \vec{e}_j = (b_{i1}\vec{s}_1 + b_{i2}\vec{s}_2)^T D^2 v (b_{j1}\vec{s}_1 + b_{j2}\vec{s}_2) \\ &= (b_{i1}b_{j2}) \begin{pmatrix} (\vec{s}_1)^T \\ (\vec{s}_2)^T \end{pmatrix} D^2 v (\vec{s}_1 \vec{s}_2) \begin{pmatrix} b_{j1} \\ b_{j2} \end{pmatrix}. \end{aligned} \quad (6.19)$$

Therefore,

$$V \equiv \left(\frac{\partial^2 v}{\partial q_i \partial q_j} \right)_{i,j=1,2} = B \left(\frac{\partial v}{\partial s_i \partial s_j} \right) B^T, \quad (6.20)$$

$$\text{where } B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}.$$

3. Perform the Euler step for system (6.13).
4. Using the matrices $\mathbf{Q}(x_0, y_0, t_{k+1})$ find $v(x_0, y_0, t_{k+1})$:

$$v(x_0, y_0, t_{k+1}) = \sqrt[4]{\det \mathbf{F}(x_0, y_0, t_{k+1}) (\det \mathbf{Q}(x_0, y_0, t_{k+1}))^2}. \quad (6.21)$$

6.2 Step 2: Recomputation of the velocity using the found image rays

The ray tracing algorithm outlined in the previous section computes the image rays significantly more accurately than it estimates the velocity $v(x_0, y_0, t_0)$. This gives us an opportunity to recompute the velocity more accurately using the found image rays. It was shown in Popov [2002] that the geometrical spreading of the rays equals $|\det \mathbf{Q}|$. We estimate it as the ratio of the areas of the grid cells at time $t = t_k$ and at $t = 0$ and then compute the velocity by formula (6.21).

6.3 Step 3: Time-to-depth conversion algorithm

The motivation and the main building block for this algorithm is Sethian's fast marching method (see Sethian [1996]). It is a 3D upgrade of the time-to-depth conversion algorithm presented in Section 3.1. In this section, we will work with the slowness s , the reciprocal of the velocity v , for convenience. Also we switch the notation for time from t_0 to T to be consistent with the notations in the Eikonal equation and Section 3.1.

We are given $s(x_0, y_0, T)$. We want to find $s(x, y, z)$, $x_0(x, y, z)$, $y_0(x, y, z)$, $T(x, y, z)$. These functions relate according to the following system of PDE's:

$$\begin{aligned} |\nabla T|^2 &= s^2(x_0(x, y, z), y_0(x, y, z), T(x, y, z)), \\ \nabla x_0 \cdot \nabla T &= 0, \\ \nabla y_0 \cdot \nabla T &= 0. \end{aligned} \tag{6.22}$$

The first equation is the Eikonal equation with an unknown right-hand-side. The other two are the orthogonality relations reflecting that the image rays are orthogonal to the equi-time surfaces. The derivation of these orthogonality relations is based on the fact that x_0 and y_0 remain unchanged along the image rays and is very similar to the derivation of the orthogonality relation (3.2) in Section 3.1.1 for the 2D case.

The numerical algorithm for solving system (6.22) is very similar to the one described

in Section 3.1.2, except for the 3-point update is added in the 3D case. The computational cost is $O(N^3 \log N)$ as for the 3D fast marching method. The equations for 1-, 2- and 3-point update are the following. Suppose we need to find s , x_0, y_0 and T at the point P .

1. 1-point-update. Let A be the a known nearest neighbor of P , and there are no known neighbors of P lying on the other grid lines. h_a can be any of h_x, h_y, h_z , depending on which grid line the points P and A lie on.

$$\frac{T(P) - T(A)}{h_a} = s(x_0(A), y_0(A), T(P)). \quad (6.23)$$

2. 2-point-update. Let A and B be two known nearest neighbors of P lying on different grid lines, and there is no known nearest neighbor lying on the other grid line. h_a and h_b can be any pair of different symbols of h_x, h_y, h_z , depending on the arrangement of the points P, A and B .

$$\begin{aligned} \frac{(T(P) - T(A))^2}{h_a^2} + \frac{(T(P) - T(B))^2}{h_b^2} &= s^2(x_0(P), y_0(P), T(P)), \\ \frac{(T(P) - T(A))(x_0(P) - x_0(A))}{h_a^2} + \frac{(T(P) - T(B))(x_0(P) - x_0(B))}{h_b^2} &= 0, \\ \frac{(T(P) - T(A))(y_0(P) - y_0(A))}{h_a^2} + \frac{(T(P) - T(B))(y_0(P) - y_0(B))}{h_b^2} &= 0, \end{aligned} \quad (6.24)$$

$$s(P) = s(x_0(P), y_0(P), T(P)),$$

$$T(P) \geq \max\{T(A), T(B)\}, \quad x_1 \leq x_0(P) \leq x_2, \quad y_1 \leq y_0(P) \leq y_2,$$

where $x_1 = \min\{x_0(A), x_0(B)\}$, $x_2 = \max\{x_0(A), x_0(B)\}$, $y_1 = \min\{y_0(A), y_0(B)\}$, $y_2 = \max\{y_0(A), y_0(B)\}$.

3. 3-point-update. Let A, B and C be three known neighbors of P all lying on different

grid lines. h_a, h_b, h_c is any permutation of h_x, h_y, h_z .

$$\begin{aligned}
& \frac{(T(P) - T(A))^2}{h_a^2} + \frac{(T(P) - T(B))^2}{h_b^2} + \frac{(T(P) - T(C))^2}{h_c^2} = s^2(x_0(P), y_0(P), T(P)), \\
& \frac{(T(P) - T(A))(x_0(P) - x_0(A))}{h_a^2} + \frac{(T(P) - T(B))(x_0(P) - x_0(B))}{h_b^2} \\
& + \frac{(T(P) - T(C))(x_0(P) - x_0(C))}{h_c^2} = 0, \\
& \frac{(T(P) - T(A))(y_0(P) - y_0(A))}{h_a^2} + \frac{(T(P) - T(B))(y_0(P) - y_0(B))}{h_b^2} \\
& + \frac{(T(P) - T(C))(y_0(P) - y_0(C))}{h_c^2} = 0.
\end{aligned} \tag{6.25}$$

$$s(P) = s(x_0(P), y_0(P), T(P)),$$

$$T(P) \geq \max\{T(A), T(B), T(C)\}, \quad x_1 \leq x_0(P) \leq x_2, \quad y_1 \leq y_0(P) \leq y_2,$$

where $x_1 = \min\{x_0(A), x_0(B), x_0(C)\}$, $x_2 = \max\{x_0(A), x_0(B), x_0(C)\}$, $y_1 = \min\{y_0(A), y_0(B), y_0(C)\}$, $y_2 = \max\{y_0(A), y_0(B), y_0(C)\}$.

Note that whenever we use the 1-point update, we artificially put the point P on the image ray passing through the point A . Whenever we use the 2-point-update, we artificially create a symmetry with respect to a plane (ABP). Taking this into account, we accept the following update rule: *2-point-update replaces 1-point-update and 3-point-update replaces 2-point-update whenever it is possible*. This algorithm has the same hypothetical causality issue as its 2D version (see Section 3.1.3). We have not encounter any causality violation in our numerical experiments. This causality issue is the subject of our future research.

In order to avoid the boundary effects (see Section 3.1.4), no image ray may enter the domain $[x_{min}, x_{max}] \times [y_{min}, y_{max}] \times [0, z_{max}]$ through the side faces. That is, the boundary image rays must be either straight or leave the domain and never re-enter.

Chapter 7

Synthetic data examples in 3D

In this section we demonstrate the ray tracing approach in 3D.

7.0.1 Example 1

Consider the following velocity field with the background velocity of 1.5 km/sec and a Gaussian anomaly centered at the depth of 2 km:

$$v(x, y, z) = 1.5 + \exp(-0.2(x^2 + y^2) - 0.3(z - 2)^2). \quad (7.1)$$

First we create the input $F(x_0, y_0, t_0)$ (see equation (6.1)), by shooting characteristics and solving system equation (6.13) in the time domain

$$x_{min} = -5 \text{ km} \leq x_0 \leq x_{max} = 5 \text{ km}, \quad y_{min} = -5 \text{ km} \leq y_0 \leq y_{max} = 5 \text{ km}, \quad (7.2)$$

$$0 \leq t_0 \leq t_{max} = 2 \text{ sec}$$

on the $50 \times 50 \times 50$ $n_x \times n_y \times n_t$ mesh. Then we apply consequently first the ray tracing algorithm to find the velocities $v(x_0, y_0, t_0)$ from the matrices $\mathbf{F}(x_0, y_0, t_0)$ and then the time-to-depth conversion algorithm to find the velocity in the depth coordinates $v(x, y, z)$ from the velocity in the time coordinates $v(x_0, y_0, t_0)$. We obtain the output in the depth

domain

$$\begin{aligned} x_{min} = -5 \text{ km} \leq x \leq x_{max} = 5 \text{ km}, \quad y_{min} = -5 \text{ km} \leq y \leq y_{max} = 5 \text{ km}, \\ 0 \leq z \leq z_{max} = 3.409 \text{ km} \end{aligned} \quad (7.3)$$

on the $50 \times 50 \times 40 \ n_x \times n_y \times n_z$ mesh.

In 2D we compared the results of our approaches with the results of the Dix inversion converted to the depth domain. The results indicate that the algorithms improve the Dix inversion and that our approaches can do qualitatively better than the Dix inversion. We would like to have something to compare the results of our ray tracing approach in 3D as well. We take the following heuristic estimate of the velocity:

$$v_{heur} = \sqrt[4]{\det \mathbf{F}}, \quad (7.4)$$

which is a 3D analog of the Dix velocity in 2D, and convert it to depth using our time-to-depth conversion algorithm.

The results are presented in Fig. 7.1. The velocity $v(x, y, z)$ is shown at the depths of 87.4 m, 262.2 m, 437.0 m, ..., 3408.6 m, the interval between slices is 174.8 m. At each depth, the dark blue color corresponds to $v = 1.5$ km/sec, and the dark red color corresponds to $v = 2.5$ km/sec. The exact velocity is shown in Fig. 7.1(a). The velocity recovered by our ray tracing approach is shown in Fig. 7.1(b). The heuristic estimate of the velocity (see. equation (7.4)) converted to depth is shown in Fig. 7.1(c). The image rays projected onto the surface are shown in Fig. 7.1(d).

First, we note that we were able to obtain a velocity estimate below the center of the Gaussian anomaly using our ray tracing approach. Second, up to the depth where the center of the anomaly lies, our results are quite accurate. Third, throughout all the depth domain, our velocity is more accurate, and in the medium depths even qualitatively more accurate than the heuristic estimate analogous to the Dix inversion.

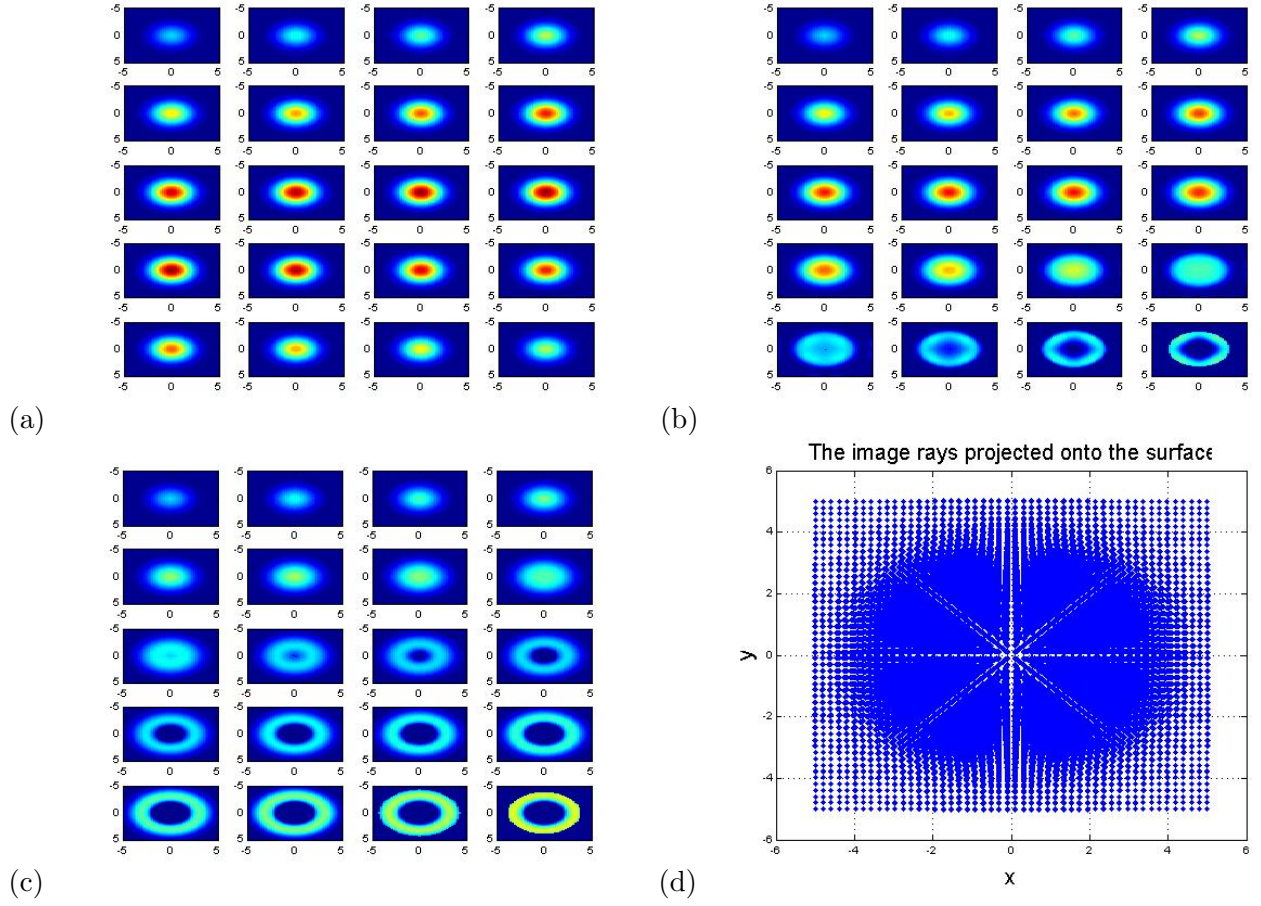


Figure 7.1. Example 1. (a) The exact velocity; (b) the velocity found by our ray tracing approach; (c) the heuristic estimate estimate analogous to the Dix inversion, converted to depth; (d) the image rays projected onto the earth surface.

7.0.2 Example 2

In this example, we also consider a velocity field with a Gaussian anomaly centered at the depth of 2 km, but with smaller variances:

$$v(x, y, z) = 1.5 + \exp(-0.5x^2 - 0.3 * y^2 - 0.3(z - 2)^2). \quad (7.5)$$

The input data are computed in the time domain

$$x_{min} = -3 \text{ km} \leq x_0 \leq x_{max} = 3 \text{ km}, \quad y_{min} = -3 \text{ km} \leq y_0 \leq y_{max} = 3 \text{ km}, \quad (7.6)$$

$$0 \leq t_0 \leq t_{max} = 2 \text{ sec}$$

on a $50 \times 50 \times 50$ $n_x \times n_y \times n_t$ mesh. The output is computed in the depth domain

$$x_{min} = -3 \text{ km} \leq x \leq x_{max} = 3 \text{ km}, \quad y_{min} = -3 \text{ km} \leq y \leq y_{max} = 3 \text{ km}, \quad (7.7)$$

$$0 \leq z \leq z_{max} = 1.6568 \text{ km}$$

on a $50 \times 50 \times 20$ $n_x \times n_y \times n_z$ mesh. The results are presented in Fig. 7.2. The velocity is shown at the depths of from 0 to $z_{max} = 1.6568$ km at every 87.2 m. At each depth, the dark blue color corresponds to $v = 1.5$ km/sec, and the dark red color corresponds to $v = 2.5$ km/sec. The exact velocity is shown in Fig. 7.2(a). The velocity recovered by our ray tracing approach is shown in Fig. 7.2(b). The heuristic estimate of the velocity (see. equation (7.4)) converted to depth is shown in Fig. 7.2(c). The image rays projected onto the surface are shown in Fig. 7.2(d).

We see that the velocity recovered by our ray tracing approach is similar to the exact velocity while the heuristic estimate is qualitatively different.

7.1 Example 3

In this example, we consider a velocity field, Gaussian in z and star shaped in (x, y) . We build this field as follows. We take a closed curve shaped as a three tip star given by

$$\alpha_0 = \phi - 0.2 \cos(3\phi); \quad (7.8)$$

$$r_0 = 1 - 0.1 \cos(3\phi).$$

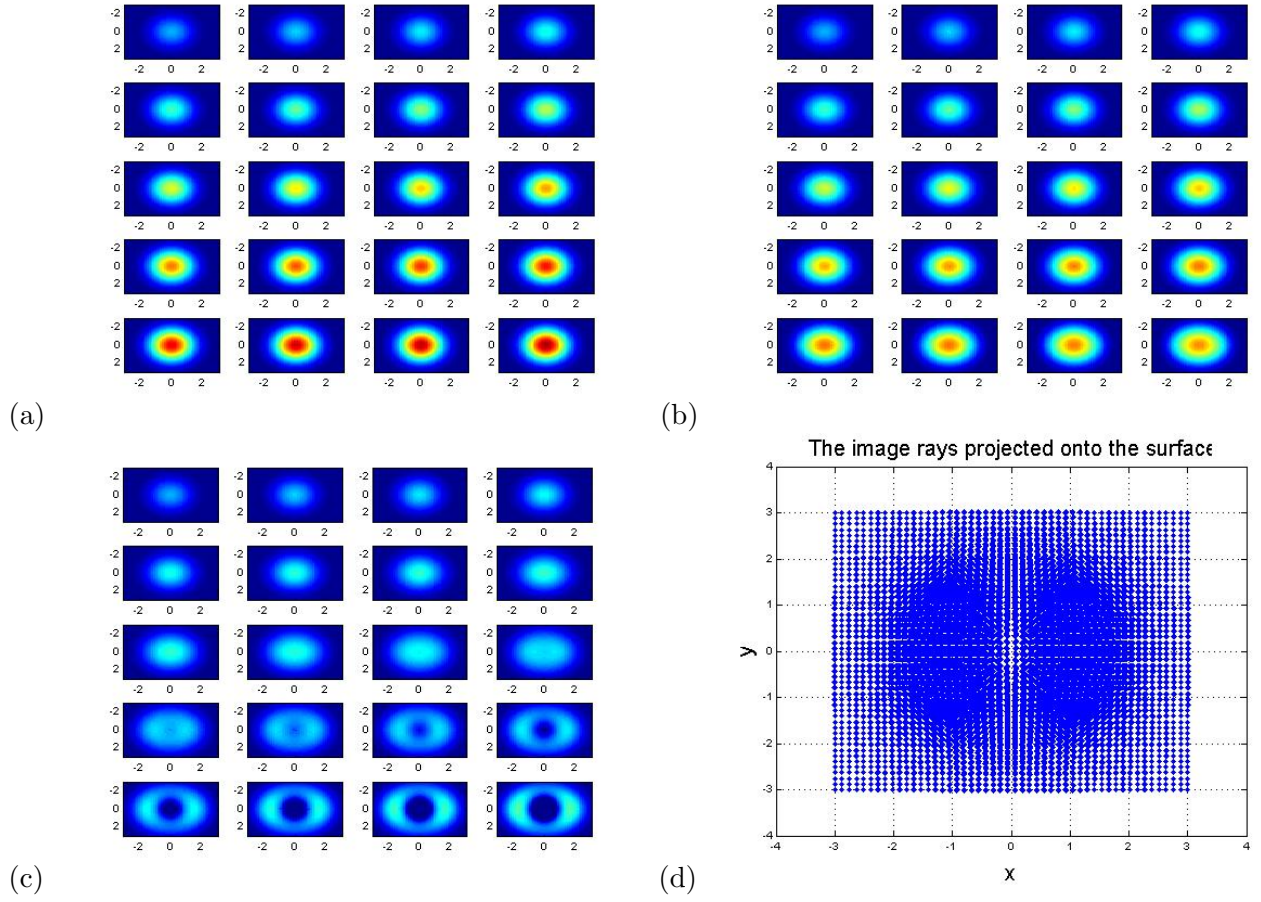


Figure 7.2. Example 2. (a) The exact velocity; (b) the velocity found by our ray tracing approach; (c) the heuristic estimate estimate analogous to the Dix inversion, converted to depth; (d) the image rays projected onto the earth surface.

Then for every point of this curve with polar coordinates (r_0, α_0) consider a spiral line defined by

$$r = r_0 + 4\sigma; \quad (7.9)$$

$$\alpha = \alpha_0 + 0.11\sigma.$$

The velocity at every point (x, y) is defined by

$$v(x, y) = 1.5 + \exp\left(-0.1 \left(\frac{d}{r_0}\right)^2\right).$$

d is the distance from the geometrical center of the level sets located at $(x_c = 0, y_c = 0)$. r_0 is the normalizing factor which is found from equations (7.8), (7.9). We equate $r = d$ and $\alpha = \arg(x, y)$ and solve for r_0 , α_0 , ϕ and σ . Then we define $v(x, y, z)$ by

$$v(x, y, z) = 1.5 + \exp\left(-0.1 \left(\frac{d}{r_0}\right)^2 - 0.5(z - 2)^2\right). \quad (7.10)$$

The input data are computed in the time domain

$$x_{min} = -8 \text{ km} \leq x_0 \leq x_{max} = 8 \text{ km}, \quad y_{min} = -8 \text{ km} \leq y_0 \leq y_{max} = 8 \text{ km}, \quad (7.11)$$

$$0 \leq t_0 \leq t_{max} = 1.5 \text{ sec}$$

on a $50 \times 50 \times 50$ $n_x \times n_y \times n_t$ mesh. The output is computed in the depth domain

$$x_{min} = -8 \text{ km} \leq x \leq x_{max} = 8 \text{ km}, \quad y_{min} = -8 \text{ km} \leq y \leq y_{max} = 8 \text{ km}, \quad (7.12)$$

$$0 \leq z \leq z_{max} = 2.027 \text{ km}$$

on a $50 \times 50 \times 32$ $n_x \times n_y \times n_z$ mesh. The results are presented in Fig. 7.3. The velocity is shown at depths of 65.4 m, 196.2 m, 327.0 m, ..., 2027 m. The interval between the slices is 130.8 m. At each depth, the dark blue color corresponds to $v = 1.5$ km/sec, and the dark red color corresponds to $v = 2.5$ km/sec. The exact velocity is shown in Fig. 7.3(a). The velocity recovered by our ray tracing approach is shown in Fig. 7.3(b). The heuristic estimate of the velocity (see. equation (7.4)) converted to depth is shown in Fig. 7.3(c). The image rays projected onto the surface are shown in Fig. 7.3(d).

The anomaly in this example has a nontrivial geometry. The velocity found by our ray tracing approach transfers the geometry qualitatively correctly, while the heuristic estimate fails to do so.

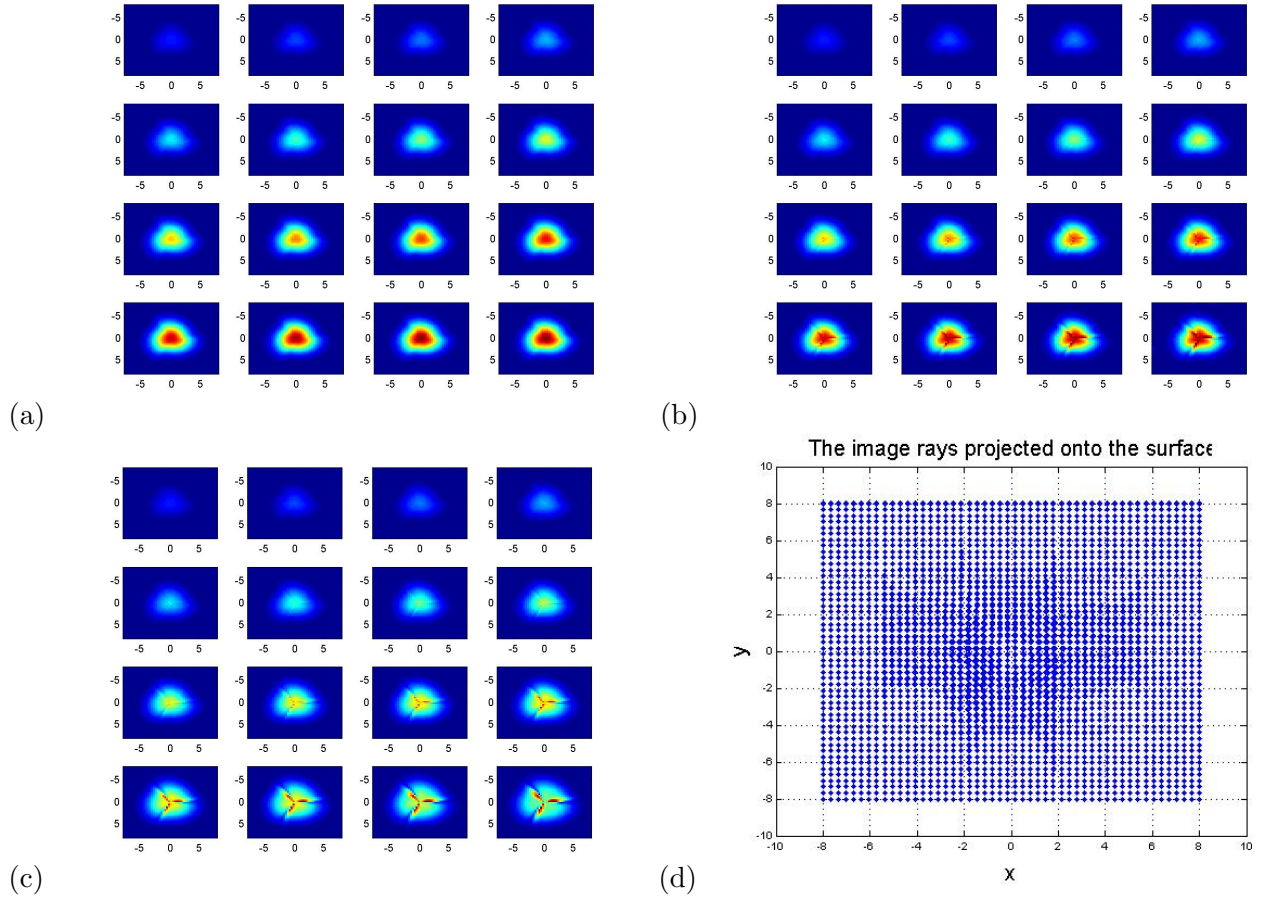


Figure 7.3. Example 3. (a) The exact velocity; (b) the velocity found by our ray tracing approach; (c) the heuristic estimate estimate analogous to the Dix inversion, converted to depth; (d) the image rays projected onto the earth surface.

Chapter 8

Summary and future work

8.1 Summary

We derived a theoretical relation between the Dix velocities in time migration coordinates and the true seismic velocities in the depth coordinates in 2D, and a relation between the matrices $\mathbf{F}(x_0, y_0, t_0)$, which can be determined from the time migration, and the true seismic velocities in the depth coordinates in 3D.

We stated the corresponding inverse problem in 2D and 3D. We showed that the inverse problem is ill-posed, i.e. sensitive to small perturbations.

However through regularized reconstruction we were able to develop numerical algorithms in 2D and 3D for obtaining smoothed seismic velocities.

We tested our algorithms on a collection of synthetic 2D and 3D examples and demonstrated that they recover the seismic velocity significantly more accurately than the Dix inversion, which is the standard estimate. Moreover, the Dix velocity may qualitatively differ from the true velocity.

We applied our level set algorithm to a field data with severe lateral velocity variation and found an estimate of the seismic velocity as well as the transition matrices from time migration coordinates to depth coordinates up to 3.3 km in depth. We used this velocity to

obtain a depth migrated image. We also used these transition matrices to convert the time migrated image to depth. This produced a good agreement between these two images. We also used the Dix velocities to produce a depth migrated image. The image obtained using the velocities estimated by our approach, and the image obtained using the Dix velocities noticeably differ in their lower part. There are several indications that the application of our approach improves the images.

8.2 Future work

8.2.1 Causality issue in the time-to-depth conversion algorithm

Our time-to-depth conversion algorithm works very fast both in 2D and 3D and produces the output in the very convenient form: on a regular rectangular mesh. However, there remains a question about a hypothetical causality problem we are asking ourselves. We devoted Section 3.1.3 to the causality issue in the time-to-depth conversion algorithm.

In 2D, we performed a large number of numerical experiments with different velocity fields trying to find an example with causality violation and did not succeed. This convinced us that the algorithm can be trusted in the application to geophysics, where the velocity fields are not very complicated in comparison with what we invented for our experiments. However, we have not proven that causality violation is impossible.

In our extension of the ray tracing approach to 3D, the hypothetical causality problem becomes more complex. We did not encounter any causality violation in 3D in our numerical experiments. However, we would like to conduct more research on this subject.

Both, in 2D and 3D, the ultimate goal is either to prove that the causality violation is impossible, or define a set of velocity fields in which the causality violation cannot happen.

8.2.2 Field data example in 3D

So far, we tested our ray tracing approach in 3D only on a collection of synthetic examples. We hope to be able to apply it to a field data. The main problem here is that we have no 3D field data now.

8.2.3 Level set approach in 3D

We are going to try to extend our level set approach for the 3D case. The main difficulty and challenge is first to modify the approach in 2D to make it narrow-band (see Adalsteinsson and Sethian [1995]) and then to extend it for 3D. This would allow to reduce significantly the computational cost. This project is important, as it would allow to enable us to estimate the velocity even if the image rays crossed, as we were able to do in 2D.

8.2.4 First and second derivatives estimation for noisy data

In our ray tracing and level set algorithms we needed to estimate the first and the second derivatives of the unknown velocity. We tried to apply many different techniques to do it. Least squares polynomial approximation using Chebyshev polynomials as a basis proved to be working the best. However, we are not completely satisfied with this solution. In 2D, the choice of the degree of the polynomial is very delicate: if the degree is too low, we oversmooth the velocity and underestimate the geometrical spreading of the image rays; if it is too high, the polynomial follows not only the main trend but also the small bumps which are the noise, and these bumps amplify due to the inheritant ill-posedness of our inverse problem (see Section 2.5.1). In 3D, we have to apply the polynomial approximation many times ($O(n_x \times n_y)$) per time step. In result we observe oversmoothing of the velocity in our numerical examples.

Bibliography

- Adalsteinsson, D. and Sethian, J.A., Fast Level Set Method for Propagating Interfaces, *J. Comp. Phys.*, **118**, 2, pp. 269-277, 1995.
- Adalsteinsson, D. and Sethian, J.A., Transport and Diffusion of Material Quantities on Propagating Interfaces via Level Set Methods, *J. Comp. Phys.*, **185**, 1, pp. 271-288, 2002.
- Červený, V., *Seismic Ray Theory*: Cambridge University Press, 2001.
- Claerbout, J., Basic Earth Imaging, <http://sepwww.stanford.edu/sep/prof/index.html>
- Dix, C. H., Seismic velocities from surface measurements: *Geophysics*, **20**, pp. 68-86, 1955
- Fomel, S., Time-migration velocity analysis by velocity continuation: *Geophysics*, **68**, pp. 1662-1672, 2003
- Hubral, P., Time migration - Some ray theoretical aspects: *Geophys. Prosp.*, **25**, pp. 738-745, 1977
- Hubral, P., Krey, T., Interval velocities from seismic reflection time measurements: SEG, 1980.
- Osher S., Sethian, J. A., Front propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations, *J. Comp. Phys.*, **79**, pp. 12-49, 1988.
- Popov, M. M., Ray theory and gaussian beam method for geophysicists: Salvador: EDUFBA, 2002.
- Popov, M. M, Pšenčík, I., Computation of ray amplitudes in inhomogeneous media with curved interfaces: *Studia Geoph. et Geod.*, **22**, pp. 248-258, 1978.
- Sethian, J. A., An Analysis of Flame Propagation: Ph.D. Dissertation, Department of Mathematics, University of California, Berkeley, CA, 1982.
- Sethian, J. A., Curvature and the Evolution of Fronts: *Commun. in Math. Phys.*, **101**, pp. 487-499, 1985.
- Sethian, J. A., Numerical methods for propagating fronts in *Variational Methods for Free Surface Interfaces*, (eds. P. Concus & R. Finn), Springer-Verlag, NY, 1987.
- Sethian, J. A., A Fast Marching Level Set Method for Monotonically Advancing Fronts: *Proceedings of the National Academy of Sciences*, **93**, 4, 1996.
- Sethian, J.A., Fast Marching Methods: *SIAM Review*, Vol. 41, No. 2, pp. 199-235, 1999.
- Sethian, J. A., *Level set methods and fast marching methods*: Cambridge University Press, 1999.
- Sethian, J. A., Vladimírsky, A., Ordered Upwind Methods for Static Hamilton-Jacobi Equations: Theory and Algorithms: *SIAM J. Numer. Anal.*, **41**, 1, pp. 325-363, 2003
- Shah, P. M., Use of wavefront curvature to relate seismic data with subsurface parameters: *Geophysics*, **38**, 812-825, 1973.
- Tsitsiklis, J.N., Efficient Algorithms for Globally Optimal Trajectories: *IEEE Tran. Automatic Control*, **40**, pp. 1528-1538, 1995.
- Yilmaz, O., *Seismic Data Analysis*: Soc. of Expl. Geophys., 2001